DAICLIC

REVUE INTERNATIONALE DES CLUBS ET UTILISATEURS DU DAI P.C.

TRIMESTRIEL
Publié par le club:

I.D.C. (International DAI Club)

I.D.C. BORDEAUX

D.A.I.C.

CAROLO DAI

MICRODAI

I.D.C.

NUMERO 2 - MAI 1985

SOMMAIRE

SOMMAIRE DU NUMERO 2 - MAI 1985

ARTICLES:	AUTEURS:	PAGE:
COUVERTURE		0
SOMMAIRE		1
EDITORIAL	IDC	2
INFORMATIONS UTILES		3
COURRIER		4
IDC BORDEAUX NEWS	IDC BORDEAUX	5
D.A.I.C. NEWS	D.A.I.C.	8
COMMENT FONCTIONNE LE MICROPROCESSEUR	Philippe RASQUIN	10
UTILITAIRE: TABLE DE FORMES	Pascal JANIN	12
BASIC: TRAMES	Félicien GILSON	17
ADAPTATION D'UN CLAVIER PROFESSIONNEL	Eric NEVE	21
LES 'GRANDES' DAICOUVERTES	Fabrice DULUINS	22
HARDCOPY MODE 0	Marc VANDERMEERSCH	24
MODES 5, 6, 5A et 6A	Stefan GOLLER	26
SAUVEGARDE DE L'ECRAN GRAPHIQUE		29
GRAPHIQUES: QUADERS		31
L'ORDINATEUR, ETALON DE FREQUENCE	Alain MARIATTE	32
JOYSTICKS ATARI	Bruno DELANNAY	36
AMD 9511	Fabien FOJUD	44
DOSSIER ASSEMBLEUR: PART.1	Henri-Pierre LEGRY	45
CARTE X-BUS	Eric CHOPPINET	51
PETITES ANNONCES		55
DAI NEWS		55
UNE SOURIS POUR VOTRE DAI	Dominique CARLIER	56
INTERFACE JOYSTICKS ATARI	Pascal JANIN	60
PROTECTION/DEPROTECTION DE PROGRAMMES	Henri-Pierre LEGRY	65
EXTENDED BASIC: LIBMAT.D	Fernand LEMOINE	69
TEST DAI DOS 1541	Marc VANDERMEERSCH	71
DAILASSEMENT	IDC	73
READ AND WRITE	Henri-Pierre LEGRY	74
STABILISATION DE L'IMAGE	DAINAMIC GERMANY	80

EDITORIAL

Trois mois maintenant que le premier numéro de DAICLIC est sorti avec pour démarrer, une soixantaine d'abonnés. Cette seconde édition est aujourd'hui envoyée à près de 200 membres; nous ne pouvons donc que vous remercier de votre confiance et de votre intéret pour notre revue qui ne cesse d'ailleurs de grossir (80 pages, c'est un record!)...

Nous espérons que ce DAICLIC plus fourni que la normale vous satisfera en tout points. Si vous avez des critiques à formuler, envoyez-les à la rédaction (c'est le seul moyen que nous avons de proposer une revue qui répond le mieux à vos souhaits). Nous vous rappellons que vos articles et programmes sont toujours les bienvenus et si nous en avions suffisamment, nous pourrions envisager la publication d'un numéro supplémentaire cette année. Vous savez donc ce qu'il vous reste à faire...

Vous remarquerez dans ce numéro, 2 articles concernant l'adaptation des joysticks ATARI au DAI. Ne vous inquiétez pas, c'est normal! Vous pourrez de cette façon, choisir celui qui vous convient le mieux... (vous pouvez aussi essayer les deux!...)

Nous tenons à féliciter Bruno DELANNAY et Gaston LAFARGUE pour le dynamisme dont ils font preuve dans le développement de leur club IDC BORDEAUX. En effet, ce club qui est le dernier en date à s'etre associé à IDC, est tout près de passer premier en nombre. Pour plus de détails concernant les activités de IDC BORDEAUX, consultez la rubrique du club dans ce numéro.

Dans le prochain numéro, nous espérons pouvoir vous présenter un nouvel interface développé par Pascal JANIN: un synthétiseur vocal de très faible cout! Encore un peu de patience... De plus, Eric NEVE a obtenu un schéma d'extension ROM et/ou RAM jusqu'à 384 K!!! L'extension comporte 2 circuits superposés et connectés au X-BUS par un cable plat de 50 conducteurs. Si vous etes amateurs, contactez-nous et nous pourrons envisager la réalisation de ces circuits à un niveau professionel.

I.D.C.

++++++++++++++++++++++++++++++++++
+++++ INFORMATIONS UTILES +++++
++++++++++++++++++++++++++++++++++

ABONNEMENTS:
L'abonnement compris dans l'inscription est valable de janvier à décembre. Un abonnement contracté pendant l'année donne droit aux numéros déjà parus dans l'année. Si une personne est inscrite à plusieurs clubs associés, les abonnements supplémentaires lui sont remboursés à raison de 500 FB pour la Belgique et 600 FB pour l'étranger (moins les frais éventuels de remboursement).

Prix de l'inscription à I.D.C.: Belgique: 750 FB
Etranger: 850 FB

INSCRIPTIONS ET ACHATS DE PROGRAMMES:
BELGIQUE: - mandat postal ou chèque
- virement sur le compte BBL: 371-0356842-45
DULUINS ET POELS
ALLEE TOUR RENARD 4
1400 NIVELLES

ETRANGER: - mandat postal international uniquement

ADRESSES:

POUR VOUS INSCRIRE OU COMMANDER DES PROGRAMMES:
IDC
C/O Fabrice DULUINS
Allée Tour Renard 4
B-1400 NIVELLES (BELGIQUE)
☎ (0)67 / 21.82.10

POUR ENVOYER VOS ARTICLES ET PROGRAMMES A PUBLIER:
IDC - REDACTION
C/O Christian POELS
Rue des Bas-Sarts 10
B-4100 SERAING (BELGIQUE)
☎ (0)41 / 37.16.06

POUR COMMERCIALISER VOS PROGRAMMES:
IDC - SOFTWARE
C/O Marc VANDERMEERSCH
Avenue du Vert Bocage 17
B-1410 WATERLOO (BELGIQUE)
☎ (0)2 / 354.13.63

CONSEIL D'ADMINISTRATION DE I.D.C. (International DAI Club):
PRESIDENT: Christian POELS, rue des Bas-Sarts 10, B-4100 SERAING
SECRETAIRE: Marc VANDERMEERSCH, avenue du Vert Bocage 17, B-1410 WATERLOO
TRESORIER: Fabrice DULUINS, allée Tour Renard 4, B-1400 NIVELLES

CONTACTS CLUBS ASSOCIES:
CAROLO-DAI: Etienne SZIGETVARI, Route Provinciale 7, B-1361 CLABECQ
DAIC: Jacques MOENS, Clos Fontaine des Ducs 6, B-1310 LA HULPE
IDC BORDEAUX: Bruno DELANNAY, Rés.Les ACACIAS B+ B3,Av.de SAIGE, F-33600 PESSAC
MICRODAI: Victor LESPINEUX, Rue Latour 39, B-4020 LIEGE

COPYRIGHT: Les articles publiés n'engagent que la responsabilité de leur auteur. Toute reproduction, meme partielle, de ce magazine est interdite sans l'accord de l'éditeur responsable.

EDITEUR RESPONSABLE: Christian POELS, rue des Bas-Sarts 10, B-4100 SERAING

COURRIER

COURRIER DES LECTEURS

J'ai lu dans le premier numéro de votre revue, que vous donniez l'adresse de la société PRODATA et que vous citiez le nom de M.ART. Je disposais déjà, pour ma part, de ces renseignements, et pour cause: j'ai en effet, pas mal de choses à leur reprocher et je voudrais à ce propos mettre les lecteurs de DAICLIC en garde.

Voici le détail des faits:
. Le 28 septembre 1984 je commandais sur le stand MULTISOFT du SICOB, qui était en fait tenu par M.ART, Directeur Commercial PRODATA, un double lecteur de disquettes 2 x 320 K sous DOS 3.0. J'obtenais à cette occasion une réduction de 10% et je versais par chèque un accompte de 1000 FF.
. Après avoir pris plusieurs fois contact avec MULTISOFT, je me suis apperçu que j'avais, en fait, passé commande directement auprès de la société belge. Début novembre, j'envoyais donc une première lettre avec accusé de réception à ces derniers pour leur demander des précisions sur les modalités de livraison.
. Le 14 novembre, mon accompte était tiré mais je n'ai jamais reçu de réponse. Tout juste un imprimé anonyme qui m'indiquait que suite à un contretemps les lecteurs ne seraient livrés qu'à la fin du mois.
. Six mois après, j'attends toujours!
. Depuis, je leur ai réécris à deux reprises, une première fois à M.ART puis une seconde fois directement au directeur de la société sans pour cela obtenir plus de succès.
Dans cette dernière lettre datée du 23 février, je leur spécifiais qu'étant donné qu'ils n'avaient pu me livrer dans les délais, je n'étais plus intéressé par leur matériel et je leur demandais de me rembourser mon accompte. Ces drives étaient en effet destinés à une utilisation professionelle et je n'ai pas pu me permettre d'attendre leur bon vouloir pour m'équiper.
. Aujourd'hui, je n'ai plus d'autres solution que d'entamer des poursuites pour obtenir gain de cause.

Tout cela pour vous dire que je trouve vraiment regrettable que 'notre cher DAI' soit commercialisé par des gens qui se préoccupent aussi peu de leur clientèle. Si tant est qu'il faille mettre cela sur le compte de la négligence! Je ne suis pas sur en tout cas que les coordonnées de personnes aussi peu recommandables aient leur place dans votre rubrique "adresses utiles".

Emmanuel BOUCHERON,
F-28 CHATEAUDUN CEDEX.

DAICLIC: Nous vous remercions de votre témoignage et nous sommes comme vous désolé que des problèmes pareils puissent survenir! Nous espérons toutefois que cette expérience n'a pas été vécue par d'autres DAIistes car si tel était le cas, n'hésitez pas à nous écrire; notre revue est ouverte à tous! Puissent les responsables de PRODATA lire ce courrier et réparer cette faute professionelle dans les plus brefs délais!

IDC BORDEAUX NEWS

I.D.C. Bordeaux

Le club I.D.C. Bordeaux ,associe depuis plusieurs mois au club I.D.C. a le plaisir d'annoncer a ses adherents que les differents services annonces dans ses imprimes :

- Logitheque a acces libre
- Bibliotheque d'interet general
- Atelier de reparation du club

sont d'ors et deja en action !!

La liste des programmes et des articles est disponible meme pour les membres d' I.D.C. ou des autres clubs associes sur simple envoi de votre support le plus usite :

- Simple K7 audio
- K7 memocom
- Disquette KEN-DOS (80 pistes)

accompagne d'une enveloppe suffisament timbree , a votre adresse , pour la reponse

D'autre part , grace a l'activite de I.D.C. Bordeaux une action des plus rentable pour ses membres a ete engagee :

- Un contact direct avec l'industriel Hollandais commercialisant le systeme de drives le plus performant existant a l'heure actuelle sur DAI , j'ai nomme le systeme KEN-DOS , permet d'effectuer des achats groupes payes en Francs FRANCAIS , livres a Bordeaux ,de lecteurs 800 K (ou si vous optez pour 2 lecteurs = 1,6 Mega Bytes !!) a des prix bien entendu inferieurs aux drives INDATA (c'est pas dur ...) mais aussi moindres que ceux rencontres dans les promotions faites par les boutiques les commercialisant en BELGIQUE (qui est pourtant moins chere que la FRANCE pour le materiel electronique !!!)

Pour vous renseigner sur les prochaines commandes du club ou sur les performances du systeme joindre le president : Mr DELANNAY (cf. infra)

Vous pouvez de plus venir voir tourner 'la bete' au club a BORDEAUX

Les adherents desirant entrer en contact avec d'autres adherents de leur region peuvent (et doivent ...) demander d'etre inscrit sur une liste prevue a cet effet . Plusieurs personnes y sont deja ... ne vous faites pas attendre , il est temps de sortir de votre isolement

Pour ceux qui auront fait les frais de l'exellent programme MAILING LIST de Mr POELS ils pouront beneficier de cette liste sous cette forme (si vous desirez acheter ce programme profitez des achats groupes du club : cela diminue les frais annexes de transfert d'argent a l'etranger)

IDC Bordeaux

Le club I.D.C. Bordeaux rappelle brievement les differents avantages procures a ses adherents :

- La LOGITHEQUE :

Elle regroupe des centaines de programmes dont l'acces est libre - NON - limite a un simple echange car , les personnes qui n'ont pas la chance d'avoir deja des programmes et donc n'ont rien a echanger ne pouront jamais rien avoir , c'est un cercle vicieux !! .

Son acces est gratuit , seul est demande le remboursement des frais d'envoi

A titre informatif y ont aussi ete inclus des programmes vendus par un autre club DAI : DAINAMIC . Ces programmes portent en commentaire la mention 'VENDU PAR DAINAMIC' car afin de ne pas nuire aux bonnes relations entre nos clubs ces programmes ne sont pas distribues

C'est dans le simple but d'informer nos membres de l'existance de ces logiciels que leur nom et leur but ont ete decrits

En effet la plupart des clubs tirant leur fond de roulement des logiciels qu'ils commercialisent il serait injuste de les en priver d'autant plus qu'il vous est possible aussi a vous de commercialiser vos oeuvres en nous les envoyant

Toujours dans le but d'aider ses adherents le club I.D.C. Bordeaux ne se limite pas a la simple information mais effectue le regroupement de vos commandes et se charge du paiement a l'etranger : vous remboursez le club en Francs FRANCAIS !!

- La BIBLIOTHEQUE :

Elle regroupe , classes selon leur Theme plusieurs dizaines de kilogrammes de documents , articles , ecrits originaux , sur votre chere machine , ce qui a represente des semaines d'effort de classement mais permet maintenant un acces tres rapide sur une somme d'information tres impressionnante , grandement facilitee par la liste decrivant le contenu general , la source d'information et le nombre de pages de chaque sujet

A la maniere de la logitheque l'acces a cette information pour nos adherents implique le remboursement des frais de photocopie et d'envois

- Un service TIRAGE DE LISTING :

Afin d'aider nos adherents n'ayant pas d'imprimante mais desirant etudier de plus pres certains programmes le club lui propose , a prix coutant , l'envoi du listing demande !!

Pour ne pas submerger le bureau une limitation a 15 programmes par mois et par adherent , ainsi que 50 photocopies , a ete decidee

Dans un prochain numero (je commence a prendre trop de place !) je vous parlerais de nos activites 'HARD' : petit atelier de reparation regroupement de personnes interessees par les interfacages les plus divers

Pour finir donc je vous rappelle que l'adhesion s'eleve a : 150 FF pour l'annee 1985 et comprend l'acces a tous nos services ainsi que l'abonnement a DAICLIC

Si par cas certains membres d' I.D.C. ou d'un club associe desirait adherer afin de profiter d'un de nos avantages , leur adhesion serait reduite , en raison de leur abonnement deja effectue a DAICLIC , a 100 FF

- Adresses des differents responsables de I.D.C. Bordeaux :

President et Tresorier :

Mr DELANNAY BRUNO
Res. Les ACACIAS B+ B3
Avenue DE SAIGE
33 600 PESSAC

Tel : (56) 45-87-70

VICE PRESIDENT ET SECRETAIRE :

Mr LAFARGUE GASTON
2 HAMEAU DE LISSANDRE
33 150 CENON

Tel : (56) 86-35-74

Responsable de La SECTION 'HARD' :

Mr PAUPERT FRANCOIS
893 ROUTE DU PEGLE
40 000 MONT DE MARSAN

Tel : (58) 75-56-83

INFOS IDC BORDEAUX

PLAYDAI

Certains disent que les revues d'informatique sont trop austères. Nous avons donc pensé pouvoir vous dérider quelques instants au cours de votre lecture en vous proposant un texte moins ardu, enfin nous l' espérons, pour vos méninges et votre intelligence (artificielle et superficielle).

Connaissez-vous par exemple celle de l' arbre (programmatique évidemment) qui arrive dans un hardware store et y rencontre sa bonne amie la stack (pas encore overflow heureusement)!?

Il lui dit: "Eh, ma vieille, barre-toi récursivement. J'ai vu dans le coin un pattern à listes, garbage collector et redresseur de tores (de ferrites) à ses heures...

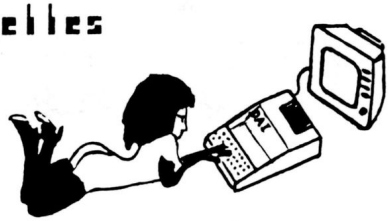
Sur ce, passe une variable (alphanumérique s.v.p.), complexée, mais très excitante, qui avait de plus complètement perdu la mémoire, (RAM si je me souviens bien).

Elle se laisse draguer par le vieux PASCAL, algolike sur les bords, et mal rafraichi en plus. Mais il lui envoya des signaux spéciaux par le bus (de données, je crois) et lui expliqua le système (d'exploitation).

Et il advint ce qui devait arriver, (c'était écrit dans son programme) et que vous avez tous deviné, du moins je l'espère. Enfin, ils vécurent très heureux et eurent beaucoup de petites puces...

Moralité : TO BIT OR NOT TO BIT, THAT IS THE QUESTION !

D.A.I.C Bruxelles



PETITE ANNONCE

L'ordinateur, qui s'est de plus en plus démocratisé au point de devenir domestique, engendre aussi le stress. Lorsque l'ordinateur refuse de coopérer, on peut désormais lui administer quelques volées de coups pour soulager ses frustrations...,sans endommager l'appareil.

Il suffit en effet pour cela de posséder un **Bit Banger**, compatible avec toutes les marques (même le DAI). Ce nouvel élément ne doit d'ailleurs pas être intégré à l'ordinateur puisqu'il est réservé à un usage externe.

De quoi s'agit-il? Tout simplement d'un maillet à tête de mousse et à manche de plastique souple. Pour Mr. Gordon Smith, Président du C.A.I.D., importateur de ce nouvel article, le **Bit Banger** exercerait également un effet thérapeutique bienfaisant sur les sujets souffrant de névrose induite par les jeux sur micro-ordinateurs...

Pour tous renseignements et prix, prière de s'adresser directement à la rédaction de la revue.

==========

JEU-CONCOURS

Ecrire un programme en BASIC pour résoudre le petit problème suivant :

Le nombre 1233 possède une propriété particulière. Si on élève chacune des deux tranches de deux chiffres de ce nombre au carré, la somme des carrés obtenus n'est autre que le nombre 1233 lui-même. O.K.?

On a bien, en effet , 12 * 12 + 33 * 33 = 144 + 1089 = 1233. Pigé cette fois?

Bon! Eh bien un autre nombre de quatre chiffres présente cette particularité. Lequel?

Envoyez vite votre réponse à Jacques MOENS, Clos Fontaine des Ducs, 6 à 1310 LA HULPE (Belgique). La première bonne réponse vaudra à son auteur de recevoir une cassette bourrée de programmes.

Ceci n'est plus une blague. Alors, tous à vos claviers!

==========

SYSTEME

COMMENT FONCTIONNE LE MICROPROCESSEUR

Un programme se compose d'une liste d'instructions. Le microprocesseur possede une petite memoire de compteur ordinal ou il stocke l'adresse memoire sur laquelle il est en train de travailler.

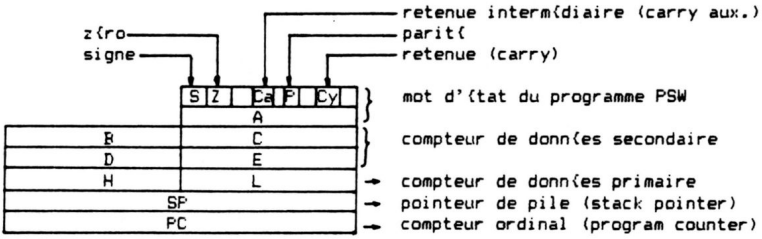
Le compteur ordinal contient l'adresse d'une case memoire. Le contenu de la case memoire est transcrit dans le microprocesseur (le contenu de la case memoire est intact) et le microprocesseur execute l'instuction.
Une autre petite memoire appelee accumulateur recoit les donnees ou les resultats avant ou apres execution de l'instruction. Au cours de l'execution, le compteur ordinal est auto-matiquement incremente et pointe l'instruction suivante.

Chaque case memoire contient 8 bits (1 octet) mais parfois le microprocesseur travaille sur 16 bit il a donc besoin de case memoires. Le compteur ordinal est alors incremente 2 fois.

La structure complete du microprocesseur est la suivante (photocopie P.186). On y notera la presence de multiples registres BCDEH et L mis a la disposition de l'utilisateur. Ce sont des registres de 8 bits chacun pouvant etre assembles par paires BC DE ou HL pour former un ensemble de 16 bits.

Le 8080 possede les instructions resumees sur les 2 pages suivantes; la premiere donne les mnemomiques le code hexa, la suivante donne le code d'instruction binaire et le temps exige par l'execution de chaque instruction.

Le schema de la page est interessant mais l'informaticien n'a pas besoin de tout les renseignements qu'il comporte. On lui preferera donc le suivant plus pratique:



L'accumulateur est la destination principale des instructions a un ou deux operandes. Une instruction doit d'abord charger l'accumulateur avant que le microprocesseur puisse effectuer une operation arithmetique ou logique. Les registres B,C,D,E,H et L sont d'un acces d'egale facilite, ils peuvent etres transferes dans n'importe quel autre registre, ou servir de deuxieme operande dans les 2 instructions a 2 operandes. L'utilisation n'est pas la meme pour tous les registres, il y a quelques petites differences entre HL et BCDE.

Les registres HL constituent le pointeur principal de donnees du 8080, c'est-a-dire que c'est dans ces 2 registres que nous placerons l'adresse de la case memoire (16 bits) a utiliser. Le progammeur devra donc autant que possible adresser les cases memoires via HL.

SYSTEME

Les registres B C D et E servent d'accumulateur secondaire, la majorite des instructions a 2 operandes font intervenir une donnee comme dans l'un de ces registres. Il y a quelques instructions pour lesquelles les registres B et C ou C et D contiennent l'adresse (16 bits) d'une donnee, mais ces instructions ne permettent que le transfert de donnee entre la memoire et l'accumulateur. Les registres B C D et E seront donc utilises pour stocker des adresses ou donnees.

Les bits de condition sont au nombre de 5, ils portent sur les resultats des operations. Tous sauf le carry auxilliaire peuvent etre testes par le programme.

Le carry (CY) est le neuvieme bit dans les operations arithmetiques. Il intervient dans la rotation. Il est mis a 0 par des instructions logiques.

Le carry auxiliaire (AC) est la retenue intermediaire, c'est-a-dire celle qui peut se produire lors d'une operation arithmetique du quartet (4 bits) de plus faible poids vers le quartet de plus en plus fort poids. Il est mis a 1 lorsqu'il y a retenue sur le 3e bit.

Le bit de parite (P) est mis a un si le contenu de l'accumulateur a la suite de n'importe quelle operation logiques ou arithmetique ou logique. Il est mis a 1 chaque fois que l'accumulateur contient un nombre negatif.

ADRESSAGE EN MEMOIRE

Pour adresser une case memoire, on peut utiliser un adressage implicite ou un adressage direct. Dans le cas de l'adressage implicite on place l'adresse de la case memoire dans les registres H et L. Cette case memoire peut etre chargee dans l'accumulateur mais aussi les autres registres secondaires. Pour l'adressage direct, il n'existe que quelques instructions. Elles operent des transferts entre accumulateur et memoire.

Philippe RASQUIN,
SAIVE (B).

UTILITAIRE : TABLES DE FORMES.

Jusqu'ici, lorsque vous vouliez faire évoluer des formes sur votre écran,vous étiez obligé soit de le faire par des DRAW, DOT, FILL, soit de coder un à un vos formes en RAM puis de les reloger en RAM vidéo par un programme adéquat. Et à moins de passer par un logiciel graphique étudié pour, ce travail était pour le moins fastidieux. Que dire de plus si l'on voulait faire déplacer sa forme point par point ou meme l'agrandir !

Fini tout ça ! Le petit utilitaire (147 octets) ci-joint règlera tous vos déboires. En effet, il permet :
- le traçage d'une forme en mode texte,graphique,4 ou 16 couleurs
- le stockage en mémoire de 127 formes au maximum
- le déplacement point par point d'une forme
- l'agrandissement de cette forme jusqu'à 255 fois.

Tout cela avec le minimum de restrictions quant à l'entrée des données nécessaires.

Vous coderez votre forme selon des ordres que vous donnerez comme à une tortue dans un LOGO, c'est-à-dire :

1 = trace et monte d'un point
2 = trace et descend d'un point
3 = trace et va à gauche d'un point
4 = trace et va à droite d'un point
5 = monte d'un point
6 = descend d'un point
7 = va à gauche d'un point
8 = va à droite d'un point
#FF fin de la forme

Exemple : pour tracer un U renersé,la forme se codera comme suit
4 - 4 - 2 - 2 - 3 - 7 - 1 - 1 - #FF
avec comme point de départ le coin supérieur gauche

Cela n'est pas très compliqué, et en combinant les 8 ordres, on peut tracer toutes les formes possibles et imaginables !

Après avoir codé vos formes, il vous faudra les rentrer en n'oubliant pas le #FF final ! Repérez la première adresse les unes à la suite des autres (en commencant par l'adresse la plus faible) à partir de #390.

Exemple : soient 4 formes dont les adresses de stockage sont :
A1=#A70 A2=#B03 A3=#B47 A4=#406
il faudra rentrer les adresses comme suit :

```
#390 : #70 } A1
       #A  }
#392 : #3  } A2
       #B  }
#394 : #47 } A3
       #B  }
#396 : #6  } A4
       #4  }
```

Il est évident que tout cela est valable quel que soit le nombre de formes que vous utiliserez à concurrence de 127 (maximal inclus).

Une fois tout cela fait, c'est le programme lui-meme qui se souviendra pour vous des adresses de vos formes. Il vous suffira de lui préciser le numéro de la forme que vous désirez...

Etudions maintenant les procédures d'affichage :

MODE TEXTE :

Adresse	Fonction	Minimal	Maximal
#386	abscisse X% de la forme	0	59
#387	poids fort X% toujours à 0 ici	0	0
#38B	ordonnée Y% de la forme	0	23
#38C	SCALE = agrandissement de la forme (normal = 1)	1	255
#38E	numéro de la forme à afficher	1	127
#38F	caractère de tracé de la forme (code ASCII) attention : 8,12 et 13 à éviter	0	255

en plus : POKE #366,#C3 : POKE #367,#6A : POKE #368,#3
puis CALLM #2F0

MODE GRAPHIQUE :

Adresse	Fonction	Minimal	Maximal
#386	poids faible abscisse X% de la forme	0	XMAX
#387	poids fort " "	0	XMAX
#38B	ordonnée Y% de la forme	0	YMAX
#38C	SCALE (normal = 1)	1	255
#38E	numéro de la forme à afficher	1	127
#38F	couleur de la forme	0	23

en plus : POKE #366,#78 : POKE #367,#EF : POKE #368,#1E
puis CALLM #2F0

Les 3 POKEs supplémentaires (#366,#367,#368) nécessaires et spécifiques à chaque mode sont à refaire seulement si l'on change de mode !
De plus, le codage d'une forme peut etre utilisé indifféremment en mode texte et/ou en mode graphique !

Les seules restrictions de codage des formes seront de ne pas les charger entre les adresses #2F0 et #390 + (n * 2), n étant le nombre de formes utilisées.

Remarques : . la taille normale de votre forme telle que vous l'avez codée correspond à SCALE = 1 (POKE #38C,1)
. si vous voulez effacer une forme :
- en mode texte : POKE #38F,32
- en mode graphique : POKE #38F,20

Voici ci dessous le programme :

```
#2F0  :F5     PUSH PSW      >SAUVE REGISTRES
       C5     PUSH BC
       D5     PUSH DE
       E5     PUSH HL
       3A8B03 LDA #38B      >Y=(#38B)
       4F     MOV C,A
```

```
        3ABF03 LDA #38F       >COULEUR=(#38F)
        47     MOV B,A
        118E03 LXI D,#38E
        3ABE03 LDA #38E       >No FORME=(#38E)
        87     ADD A
        6F     MOV L,A
        2600   MVI H,0
        19     DAD D
        220B03 SHLD #30B
#30A    :2A9003 LHLD #390     >TABLE depuis #390
        EB     XCHG
        3AB603 LDA #386       >XB=(#386)
        6F     MOV L,A
        3AB703 LDA #387       >XH=(#387)
        67     MOV H,A
#316    :3ABC03 LDA #38C      >SCALE=(#38C)
        328D03 STA #38D       >REGISTRE INTERMEDIAIRE
        1A     LDAX D
        13     INX D
        F5     PUSH PSW
#31F    :FE01  CPI 1          >1=MONTE+TRACE
        CA4A03 JZ #34A
        FE02   CPI 2          >2=DESCEND+TRACE
        CA5103 JZ #351
        FE03   CPI 3          >3=GAUCHE+TRACE
        CA5803 JZ #358
        FE04   CPI 4          >4=DROITE+TRACE
        CA5F03 JZ #35F
        FE05   CPI 5          >5=MONTE
        CA4D03 JZ #34D
        FE06   CPI 6          >6=DESCEND
        CA5403 JZ #354
        FE07   CPI 7          >7=GAUCHE
        CA5B03 JZ #35B
        FE08   CPI 8          >8=DROITE
        CA6203 JZ #362
        C34CC1 JMP #C14C      >sinon:FIN
#34A    :CD6603 CALL #366     >1
        0C     INR C          >5
        C37103 JMP #371
#351    :CD6603 CALL #366     >2
        0D     DCR C          >6
        C37103 JMP #371
#358    :CD6603 CALL #366     >3
        2B     DCX H          >7
        C37103 JMP #371
#35F    :CD6603 CALL #366     >4
        23     INX H          >8
        C37103 JMP #371
#366    :78    MOV A,B        >GRAPH SUBROUT
        EF     RST 5
        1E     DATA #1E
        C9     RET
#36A    :61    MOV H,C        >TEXT SUBROUT
        EF     RST 5
        09     DATA 9
        78     MOV A,B
        EF     RST 5
        03     DATA 3
        C9     RET
#371    :3A8D03 LDA #38D
```

```
        3D      DCR A
        CA8003  JZ #37F         >SI FIN BOUCLE:NEXT DATA
        328D03  STA #38D
        F1      POP PSW
        F5      PUSH PSW
        C31F03  JMP #31F        >SUITE BOUCLE
#380  :F1       POP PSW
        C31603  JMP #316        >NEXT DATA
```

```
"D2F0 383

02F0 F5 C5 D5 E5 3A 8B 03 4F 3A 8F 03 47 11 8E 03 3A
0300 8E 03 87 6F 26 00 19 22 0B 03 2A 90 03 EB 3A 86
0310 03 6F 3A 87 03 67 3A 8C 03 32 8D 03 1A 13 F5 FE
0320 01 CA 4A 03 FE 02 CA 51 03 FE 03 CA 58 03 FE 04
0330 CA 5F 03 FE 05 CA 4D 03 FE 06 CA 54 03 FE 07 CA
0340 5B 03 FE 08 CA 62 03 C3 4C C1 CD 66 03 0C C3 71
0350 03 CD 66 03 0D C3 71 03 CD 66 03 2B C3 71 03 CD
0360 66 03 23 C3 71 03 78 EF 1E C9 61 EF 09 78 EF 03
0370 C9 3A 8D 03 3D CA 80 03 32 8D 03 F1 F5 C3 1F 03
0380 F1 C3 16 03
```

PAGE 01 -- TABLES DE FORMES DEMO

```
10      REM DEMO
11      COLORG 0 1 3 5
20      IF PEEK(#2F0)<>#F5 THEN PRINT "LE MLP N'EST PAS EN RAM !":STOP
21      POKE #366,#78:POKE #367,#EF:POKE #368,#1E
30      POKE #38E,1:REM 1 FORME
31      POKE #390,#A0:POKE #391,#3:REM STARTADRESSE FORME 1
32      MODE 2:GOSUB 1000
33      GOSUB 2000
34      GOSUB 500
35      MODE 4
36      GOSUB 500
37      MODE 6
38      GOSUB 500

39      REM TOUT CE QUI EST FAIT PRECEDEMMENT
40      REM EN MODE 4 COULEURS
41      REM PEUT ETRE FAIT EN MODE 16 COULEURS !!!
42      REM ET MEME EN MODE TEXTE: VOYEZ LA SUITE ...
43      POKE #366,#C3:POKE #367,#6A:POKE #368,3
44      MODE 0:GOSUB 1000:POKE #38C,1
45      FOR I%=0 TO 40
46    1  POKE #38F,#20:REM SPACE POUR EFFACEMENT
47    1  CALLM #2F0
48    1  POKE #386,I%+8:POKE #38F,I%+65
49    1  CALLM #2F0
50    1  WAIT TIME 10
60      NEXT
70      FOR J%=0 TO 13:POKE #38F,#20:CALLM #2F0
71      POKE #38B,J%:POKE #38F,41+J%:WAIT TIME 10:CALLM #2F0:NEXT
72      POKE #38F,#20:CALLM #2F0
73      GOSUB 1000
74      POKE #38C,2:POKE #386,16:POKE #38F,29:CALLM #2F0:POKE #38C,1:CALLM #2F0
75      CURSOR 0,23:END
```

```
323     REM #38B = Y
324     REM #386-#387 = X
325     REM #38C = SCALE (MIN=1)
326     REM #38F = COULEUR
327     REM #38E = No FORME (MIN=1)
500     FOR I%=1 TO INT((XMAX-8.0)/19.0)
501   1  POKE #38C,I%:POKE #386,8*I%:POKE #38F,21+(I% MOD 3):CALLM #2F0
502     NEXT
503     WAIT TIME 100
504     RETURN
1000    POKE #386,8:POKE #387,0
1010    POKE #38B,0:RETURN
2000    RESTORE
2005    C%=#3A0
2010    READ A%:IF A%<0 THEN POKE C%,#FF:RETURN
2011    POKE C%,A%:C%=C%+1:GOTO 2010
2019    DATA 3,3,3,3,3,3,3,3
2020    DATA 1,4,1,4,1,4,1,4,1,4,1,4,1,4,1,4
2021    DATA 4,2,4,2,4,2,4,2,4,2,4,2,4,2,4,2
2022    DATA 3,3,3,3,3,3,3,3,3,3
2023    DATA -1
```

Pour vous familiariser avec les adresses, voici un petit programme de démonstration en BASIC.

Tapez le programme DEMO et sauvez le sur une cassette ou une diskette, ensuite suivez la procédure décrite ci-dessous.

```
Procédure de chargement : *UT <return>
                          >Z3 <return>
   pour déplacer les -->  >S29B xx-00 yy-30 <curseur gauche>
    HEAP pointeurs        >GDEB8
                          >R   (charge TABLES DE FORMES LM)
                          >B
                          *LOAD (charge DEMO)
```

Pascal JANIN

Vous vous souvenez sans doute du programme BALONS du premier DAICLIC ? ... Mais si, page 41 ...
Pour ceux qui ont eu le courage de rentrer les 18 lignes qui le composaient, ils ont pu admirer la qualité de cette petite démonstration graphique... oui mais ! L'affichage des BALONS est quand meme un rien trop lent, non ?

Pour y remédier, tapez en BASIC :

```
. 150    FOR I = 0 TO R STEP 2
. 151    (supprimer la ligne)
. 152    A = SQR ( I * ( 2 * R - I ))
. faites  RUN ...
```

et comparez la rapidité !!!

Le Turbo DAIiste, Pascal JANIN

BASIC

```
10      REM .                   Surfaces
15      REM
20      REM .                   Jean-Paul     Delahaye
25      REM .                   arr.pour DAI F.Gilson
27      REM .                   ........ Fevrier 1985
28      REM
29      PRINT CHR$(12):COLORT 5 15 0 0:MODE 0:CLEAR 3000
30      FOR I%=0 TO 21:CURSOR 18+I%,15:PRINT CHR$(29):NEXT
31      FOR I%=1 TO 3
32      PRINT TAB(18);CHR$(29);TAB(39);CHR$(29):NEXT
33      FOR I%=0 TO 21:CURSOR 18+I%,11:PRINT CHR$(29):NEXT
35      CURSOR 21,13:PRINT "TRAMES EN FOLIES"
40      CURSOR 25,1:PRINT "Un programme de Jean-Paul Delahaye"
45      PRINT TAB(25);"Arrange  pour  DAI  par  F. Gilson";
47      WAIT TIME 150:PRINT CHR$(12)
50      CURSOR 11,15:PRINT "Vous pouvez choisir parmi quelques realisations"
52      PRINT TAB(11);"au choix de 1 a 8."
54      PRINT :PRINT TAB(11);"Si  vous  voulez  creer  des  figures, tapez 9"
55      INPUT "            Selection ";SEL%
60      COLORG 0 14 13 10:MODE 6
65      N%=0:PA=0.0:E1=0.0:E2=0.0
70      XA=0.0:YA=0.0:XB=0.0:YB=0.0:XC=0.0:YC=0.0:XD=0.0:YD=0.0
75      ON SEL% GOTO 80,90,100,110,120,130,140,150,6000,160
80      N%=6:PA=24.0:E1=2.0:E2=1.0:COL%=14
85      XA=190.0:YA=24.0:XB=332.0:YB=95.0:XC=190.0:YC=238.0:XD=40.0:YD=160.0:
        GOTO 200
90      N%=20:PA=3.0:E1=2.0:E2=0.0:COL%=13
95      XA=112.0:XB=334.0:YB=70.0:XC=224.0:YC=210.0:YD=140.0:GOTO 200
100     N%=42:PA=2.5:E1=2.0:E2=0.0:COL%=10
105     XA=160.0:XB=320.0:YB=120.0:XC=180.0:YC=240.0:YD=140.0:GOTO 200
110     N%=52:PA=2.0:E1=2.0:E2=0.0:COL%=14
115     XA=168.0:XB=335.0:YB=112.0:XC=168.0:YC=224.0:YD=112.0:GOTO 200
120     N%=20:PA=3.0:E1=2.0:E2=0.0:COL%=13
125     XA=126.0:XB=252.0:YB=46.0:XC=188.0:YC=180.0:YD=80.0:GOTO 200
130     N%=36:PA=2.0:E1=1.0:E2=0.0:COL%=10
135     XA=114.0:XB=330.0:XC=220.0:YC=116.0:YD=116.0:GOTO 200
140     N%=20:PA=2.0:E1=2.0:E2=1.0:COL%=14
145     XA=200.0:YA=0.0:XB=300.0:YB=10.0:XC=220.0:YC=100.0:XD=20.0:YD=70.0:GOTO
        200
150     N%=19:PA=2.0:E1=2.0:E2=0.0:COL%=13
155     XA=168.0:XB=332.0:YB=130.0:XC=168.0:YC=210.0:YD=130.0:GOTO 200

160     REM N%=  :PA=  :E1=  :E2=
165     REM XA=  :YA=  :XB=  :YB=  :XC=  :YC=  :XD=  :YD=
199     REM
200     M%=336/PA:DIM MA%(M%),MI%(M%)
300     FOR I%=0 TO M%:MA%(I%)=-1000:MI%(I%)=10
400     NEXT
500     FOR I=0.0 TO N%
510     XP=(I*XD+(N%-I)*XA)/N%
520     YP=(I*YD+(N%-I)*YA)/N%
530     XQ=(I*XC+(N%-I)*XB)/N%
540     YQ=(I*YC+(N%-I)*YB)/N%
550     IF E3=1.0 THEN Y=I/N%:GOTO 560
555     X=I/N%
560     I1=INT(XP/PA):I2=INT(XQ/PA):G=SGN(I2-I1)
600     FOR J=I1 TO I2 STEP G
610     IF E3=1.0 THEN X=(J-I1)/(I2-I1):GOTO 620
615     Y=(J-I1)/(I2-I1)
620     ON SEL% GOSUB 5000,5100,5200,5300,5400,5500,5600,5700,1,5800
```

```
630     XF=INT(J*PA)
640     YF=((J-I1)*YQ+(I2-J)*YP)/(I2-I1)+Z:YF=INT(YF)
700     IF J=I1 THEN X1%=XF:Y1%=YF
710     IF E2=1 THEN 760
720     IF YF>MI%(J) AND YF<MA%(J) THEN X1%=XF:Y1%=YF:GOTO 800
740     IF YF>MA%(J) THEN MA%(J)=YF
750     IF YF<MI%(J) THEN MI%(J)=YF
760     DRAW X1%,Y1% XF,YF COL%:X1%=XF:Y1%=YF
800     NEXT
900     NEXT
1000    IF E1=1.0 THEN 8000
1010    E3=1.0:E1=1.0
1020    UU=XD:XD=XB:XB=UU
1030    UU=YD:YD=YB:YB=UU
1050    GOTO 300
5000    Z=-80.0*SIN(2.0*PI*Y)*SIN(PI*X)
5090    RETURN

5100    REM Z=F(X,Y)
5110    X7=2.0*X-1.0:Y7=2.0*Y-1.0
5120    DI=X7*X7+Y7*Y7
5130    IF DI<1.0 THEN Z=SQR(1.0-DI):GOTO 5150
5140    Z=0.0
5150    Z=100.0*Z
5190    RETURN
5200    Z=68.0*SIN(2.0*PI*Y)*SIN(3.0*PI*X)
5290    RETURN
5300    DI=16.0*((X-0.5)*(X-0.5)+(Y-0.5)*(Y-0.5))
5310    Z=140.0*COS(4.0*DI)*EXP(-DI)
5390    RETURN
5400    X7=2.0*X-1.0:Y7=2.0*Y-1.0
5410    AA=SQR(1.0-X7*X7)*(ABS(Y7)+0.5):BB=0.7*SQR(1.0-Y7*Y7)*(ABS(X7)+0.5)
5420    IF BB>AA THEN AA=BB
5430    Z=64.0*AA
5490    RETURN
5500    X7=X:Y7=Y:K7=0.0
5510    K7=K7+1.0
5520    IF K7>5.0 THEN Z=0.0:RETURN
5530    U7=INT(3.0*X7):X7=3.0*X7-U7
5540    V7=INT(3.0*Y7):Y7=3.0*Y7-V7
5550    IF U7<>1.0 OR V7<>1.0 THEN 5510
5560    Z1=0.5-ABS(X7-0.5):Z=0.5-ABS(Y7-0.5)
5570    IF Z1<Z THEN Z=Z1
5580    Z=(900.0/3.0^K7)*Z
5590    RETURN
5600    X7=2.0*ABS(X-0.5):Y7=2.0*ABS(Y-0.5):M7=X7+Y7
5610    M8=INT(4.0*M7):M9=4.0*M7-M8
5620    Z=0.0:IF M9>0.8 THEN Z=(M9-0.8)*5.0
5630    Z=17.0*(Z+M8)
5690    RETURN
5700    DI=8.0*SQR((X-0.5)*(X-0.5)+(Y-0.5)*(Y-0.5)):Z=50.0*COS(DI)
5790    RETURN

5800    REM Sous-programme de definition de la fonction
5810    REM Z=F(X,Y)                pour vos programmes
5999    RETURN
6000    PRINT CHR$(12);:COLORT 9 14 0 0:COLORG 9 14 10 11:MODE 6A
6010    COL%=11:CO1%=10:CO2%=14:PAS1%=5:PAS2%=10
6020    DRAW 60,50 280,50 COL%:DRAW 80,30 80,200 COL%
6030    X1%=180-PAS2%:Y1%=60-PAS1%:XX1%=100-PAS2%:YY1%=140-PAS1%
```

```
6040    X2%=180+PAS2%:Y2%=60-PAS2%:XX2%=260+PAS2%:YY2%=100-PAS2%
6050    FOR I%=1 TO 9:X1%=X1%+PAS2%:Y1%=Y1%+PAS1%:XX1%=XX1%+PAS2%:YY1%=YY1%+
        PAS1%
6060    DRAW X1%,Y1% XX1%,YY1% COL%:NEXT
6070    FOR I%=1 TO 9:X2%=X2%-PAS2%:XX2%=XX2%-PAS2%:Y2%=Y2%+PAS2%:YY2%=YY2%+
        PAS2%
6080    DRAW X2%,Y2% XX2%,YY2% COL%:NEXT
6090    X%=76:X1%=78:PAS3%=5:Y%=180:FOR I%=1 TO 20:X%=X%+PAS3%:X1%=X1%+PAS3%
6100    DRAW X%,Y% X1%,Y% COL%:NEXT
6110    X%=76:X1%=78:Y%=140:FOR I%=1 TO 4:X%=X%+PAS3%:X1%=X1%+PAS3%
6120    DRAW X%,Y% X1%,Y% COL%:NEXT
6130    X%=76:X1%=78:Y%=100:FOR I%=1 TO 36:X%=X%+PAS3%:X1%=X1%+PAS3%
6140    DRAW X%,Y% X1%,Y% COL%:NEXT
6150    X%=76:X1%=78:Y%=60:FOR I%=1 TO 20:X%=X%+PAS3%:X1%=X1%+PAS3%
6160    DRAW X%,Y% X1%,Y% COL%:NEXT
6170    Y%=46:Y1%=48:X%=100:FOR I%=1 TO 18:Y%=Y%+PAS3%:Y1%=Y1%+PAS3%
6180    DRAW X%,Y% X%,Y1% COL%:NEXT
6190    Y%=46:Y1%=48:X%=180:FOR I%=1 TO 2:Y%=Y%+PAS3%:Y1%=Y1%+PAS3%
6200    DRAW X%,Y% X%,Y1% COL%:NEXT
6210    Y%=46:Y1%=48:X%=260:FOR I%=1 TO 10:Y%=Y%+PAS3%:Y1%=Y1%+PAS3%
6220    DRAW X%,Y% X%,Y1% COL%:NEXT
6230    X%=73:Y%=180:GOSUB 6260:X%=73:Y%=140:GOSUB 6260:X%=73:Y%=100:GOSUB 6260
6240    X%=73:Y%=60:GOSUB 6260:X%=73:Y%=40:GOSUB 6260:X%=101:Y%=40:GOSUB 6260
6250    X%=185:Y%=40:GOSUB 6260:X%=258:Y%=40:GOSUB 6260:X%=265:Y%=40:GOSUB 6260:
        GOTO 6270
6260    DRAW X%+1,Y% X%+3,Y% CO1%:DRAW X%+1,Y%+6 X%+3,Y%+6 CO1%:DRAW X%,Y%+1 X%,
        Y%+5 CO1%:DRAW X%+4,Y%+1 X%+4,Y%+5 CO1%:RETURN
6270    X%=68:Y%=180:GOSUB 6300:X%=61:Y%=140:GOSUB 6300:X%=173:Y%=40:GOSUB 6300:
        GOTO 6310
6300    DRAW X%,Y% X%+2,Y% CO1%:DRAW X%+1,Y% X%+1,Y%+6 CO1%:DOT X%,Y%+5 CO1%:
        RETURN
6310    X%=61:Y%=180:GOSUB 6400:X%=66:Y%=60:GOSUB 6400:GOTO 6410
6400    DRAW X%,Y% X%+4,Y% CO1%:DRAW X%+1,Y%+1 X%+4,Y%+4 CO1%:DRAW X%+1,Y%+6 X%+
        3,Y%+6 CO1%:DOT X%,Y%+5 CO1%:DOT X%+4,Y%+5 CO1%:RETURN
6410    X%=251:Y%=40:DRAW X%,Y% X%+3,Y% CO1%:DRAW X%,Y%+6 X%+3,Y%+6 CO1%:DRAW
        X%+1,Y%+3 X%+4,Y%+6 CO1%
6420    DRAW X%+1,Y%+3 X%+3,Y%+3 CO1%:DRAW X%+4,Y%+1 X%+4,Y%+2 CO1%
6430    X%=94:Y%=40:DRAW X%,Y%+2 X%+4,Y%+2 CO1%:DRAW X%+3,Y% X%+3,Y%+6 CO1%:
        DRAW X%,Y%+3 X%+2,Y%+5 CO1%
6440    X%=66:Y%=140:DRAW X%+1,Y% X%+3,Y% CO1%:DRAW X%,Y%+4 X%+3,Y%+4 CO1%:DRAW
        X%,Y%+6 X%+4,Y%+6 CO1%
6450    DRAW X%+4,Y%+1 X%+4,Y%+3 CO1%:DOT X%,Y%+1 CO1%:DOT X%,Y%+5 CO1%
6460    X%=178:Y%=40:DRAW X%+2,Y% X%+2,Y%+2 CO1%:DRAW X%+2,Y%+3 X%+4,Y%+5 CO1%:
        DRAW X%,Y%+6 X%+4,Y%+6 CO1%
6470    X%=66:Y%=100:DRAW X%+1,Y% X%+3,Y% CO1%:DRAW X%+1,Y%+3 X%+3,Y%+3 CO1%:
        DRAW X%+1,Y%+6 X%+3,Y%+6 CO1%
6480    DRAW X%,Y%+1 X%,Y%+2 CO1%:DRAW X%,Y%+4 X%,Y%+5 CO1%:DRAW X%+4,Y%+1 X%+4,
        Y%+2 CO1%:DRAW X%+4,Y%+4 X%+4,Y%+5 CO1%
6500    X%=185:Y%=52:DRAW X%,Y% X%,Y%+5 CO2%:DRAW X%+4,Y% X%+4,Y%+5 CO2%
6510    DRAW X%+1,Y%+2 X%+3,Y%+2 CO2%:DRAW X%+1,Y%+6 X%+3,Y%+6 CO2%
6520    X%=263:Y%=100:DRAW X%,Y% X%+3,Y% CO2%:DRAW X%,Y%+6 X%+3,Y%+6 CO2%:DRAW
        X%+2,Y%+3 X%+3,Y%+3 CO2%
6530    DRAW X%+1,Y%+1 X%+1,Y%+5 CO2%:DRAW X%+4,Y%+1 X%+4,Y%+2 CO2%:DRAW X%+4,
        Y%+4 X%+4,Y%+5 CO2%
6540    X%=183:Y%=180:DRAW X%,Y%+1 X%,Y%+5 CO2%:DRAW X%+1,Y% X%+3,Y% CO2%:DRAW
        X%+1,Y%+6 X%+3,Y%+6 CO2%
6550    DOT X%+4,Y%+1 CO2%:DOT X%+4,Y%+5 CO2%
6560    X%=98:Y%=145:DRAW X%,Y% X%+3,Y% CO2%:DRAW X%,Y%+6 X%+3,Y%+6 CO2%
6570    DRAW X%+1,Y%+1 X%+1,Y%+5 CO2%:DRAW X%+4,Y%+1 X%+4,Y%+5 CO2%
6580    X%=52:Y%=47:DRAW X%,Y%+1 X%+4,Y%+5 CO2%:DRAW X%,Y%+5 X%+4,Y%+1 CO2%
```

```
6590    DOT X%,Y% CO2%:DOT X%+4,Y% CO2%:DOT X%,Y%+6 CO2%:DOT X%+4,Y%+6 CO2%
6600    X%=78:Y%=20:DRAW X%+2,Y% X%+2,Y%+2 CO2%:DRAW X%,Y%+5 X%+2,Y%+3 CO2%:
        DRAW X%+4,Y%+5 X%+2,Y%+3 CO2%
6610    DOT X%,Y%+6 CO2%:DOT X%+4,Y%+6 CO2%
6700    CURSOR 32,3:PRINT "XA = 170  YA =  20"
6710    CURSOR 5,2:PRINT "Valeurs des coordonnees    XB = 300  YB =  80"
6720    PRINT TAB(5);"de l'exemple represente    XC = 170  YC = 210"
6730    PRINT TAB(32);"XD =  40  YD = 150       SPC";
7000    CALLM #D6DA
7005    PRINT CHR$(12);:PRINT TAB(10);"A la ligne 160 les variables a utiliser
        sont"
7010    PRINT :PRINT TAB(3);"N% ==> Nombre de lignes desirees (exemples 10
        20....60)"
7020    PRINT TAB(3);"PA ==> Finesse du dessin de 2 a 20 (2 = fin, mais lent) ";
7025    CALLM #D6DA:PRINT CHR$(12)
7030    PRINT TAB(3);"E1 ==> Lignes en paralleles ==>1    en quadrillage ==>2"
7035    PRINT TAB(3);"E2 ==> Les parties cachees sont dessinees =>1 sinon =>0"
7040    CALLM #D6DA:PRINT CHR$(12);
7045    PRINT TAB(7);"A la ligne 165 les variables a utiliser sont..."
7050    PRINT TAB(7);"(XA)  (XB)  (XC)  (XD)   (YA)  (YB)  (YC)  (YD)"
7060    PRINT TAB(7);"Il n'est pas necessaire  d'utiliser toutes  les"
7065    PRINT TAB(7);"coordonnees, referez vous aux exemples de 1 a 8";
7070    CALLM #D6DA:PRINT CHR$(12);
7080    PRINT TAB(7);"Placez maintenant votre formule ligne 5800 a 5990."
7100    PRINT TAB(7);"Pour faire touner votre programme,"
7110    PRINT TAB(7);"RUN :ensuite repondez a Selection ? par 10"
7115    PRINT TAB(7);"                          -----------     --";
7120    CALLM #D6DA:MODE 0:PRINT CHR$(12)
7130    CURSOR 23,12:PRINT "BON AMUSEMENT"
7140    CURSOR 40,3:PRINT "EDIT 160-165"
7150    PRINT TAB(40);"EDIT 5800-5990"
7160    END
8000    X%=10:Y%=0:DRAW X%,Y% X%,Y%+6 COL%:DRAW X%,Y%+3 X%+2,Y%+3 COL%:DRAW X%,
        Y%+6 X%+4,Y%+6 COL%:DRAW X%+1,Y% X%+4,Y% COL%
8010    X%=17:DRAW X%,Y% X%,Y%+6 COL%:DRAW X%+4,Y% X%+4,Y%+6 COL%:DRAW X%+1,Y%+
        5 X%+4,Y%+1 COL%
8020    X%=24:DRAW X%,Y% X%+3,Y% COL%:DRAW X%,Y%+6 X%+3,Y%+6 COL%:DRAW X%+1,Y%+
        1 X%+1,Y%+5 COL%:DRAW X%+4,Y%+1 X%+4,Y%+5 COL%
8500    GOTO 8500
```

HARDWARE : Adaptation d'un clavier professionnel pour le DAI.

ERIC NEVE ''S.EL.S.'' pour D.A.I.C. 25.04.1985
25 Hoogstraat 1512 DWORP
Tél. (0)2 / 380.35.81

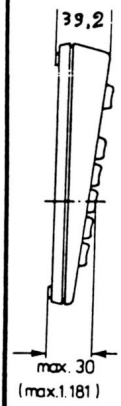
Enfin une touche professionnelle et un rajeunissement de notre DAI préféré. Après de multiples consultations de documentations, j'ai choisi un beau Clavier Professionnel dé(at)tachable comportant 87 touches (eh oui, 87 !), de la marque connue : PREH.

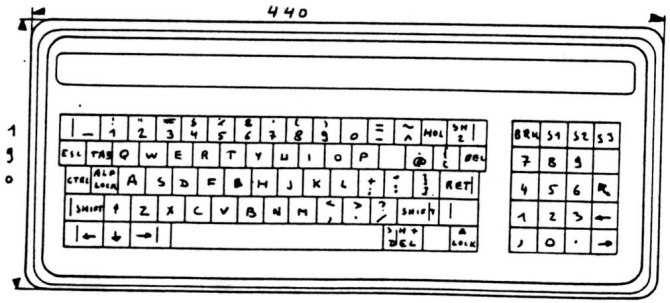
Il est encodé 8 Bits //+1 Strobe, avec répétition automatique à l'appui prolongé.
Il est commandé par un processeur du type 68705 P3 (MOTOROLA). La frappe est si agréable que l'on perd très vite l'habitude du clavier original ! Naturellement, entre ce bel engin et le DAI,il faut un Schtroumpfeur qui les relie l'un à l'autre, c'est une interface qui interprète les codes ASCII et les adapte à l'entrée du DAI,en simulant électroniquement l'appui sur les touches. Les codes de conversion (environ 185) se trouvent logés dans une Eprom. Une autre petite interface plaçée dans le clavier prolonge l'impulsion de strobe.
Sur les 87 touches 80 sont actives, il en reste donc 7 pour une programmation ultérieure de mots, par exemple IF,SPC,GOTO,GOSUB, DIM, etc,mais cela nécessite une interface à rajouter au premier, ce sera un deuxième step.
J'ai également ajouté un RESET, branché un deuxième shift direct, le HOLD du processeur, une touche pour SH+DEL à utiliser avec 1A, 1B, 1C, 1D, etc pour utiliser directement les caractères accentués avec le traitement de texte FWP.
L'autorépétition des touches est moins rapide que celle du DAI !

En changeant les touches de place et en modifiant l'Eprom de l'interface,il est facile d'obtenir un AZERTY. Avis aux amateurs!

Comme mon programmeur d'Eproms était chaud, j'en ai profité pour changer la Rom des caractères par une Eprom de 4Kb pourvue de 2 jeux de caractères à mettre à l'écran :
le € (94);é (123);è (125);ç (92);à (64) et ù (124).
Un SWITCH permet la sélection du jeux de caractères,original/new.

Le prix brut du clavier est en principe 7.800 FB + 19% de TVA ce qui est un prix unitaire mais il y a moyen de baisser les prix en groupant les commandes et l'interface + cables + connecteurs coutent environ 1.500 FB (également prix unitaire).





Les 'Grandes' DAIcouvertes.

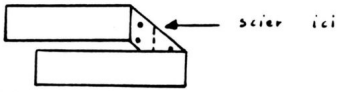
Tout DAIiste a déja été confronté, souvent après de longues heures d'utilisation, à des problèmes de surchauffe entrainant des désagréments divers voire un blocage du DAI, comble de tout cela arrive toujours au mauvais moment.
La solution décrite ci-dessous vous aidera à résoudre ces problèmes.

La solution qui m'a été suggérée par Alain Mariatte, consiste en l'installation d'un ventilateur de 'petite' taille à l'intérieur du DAI sous la grille d'aération. Ce ventilateur (voir les caractéristiques ci-dessous) est particulièrement silencieux, il coute 121 FF (prix mi-février).

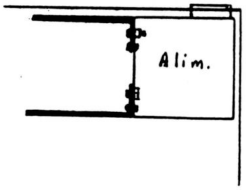
Caractéristiques :	Nom	:	ETRI
.....	Modèle	:	99 XM
	Numéro	:	99 XM 01 81 /000/ D
	Alimentation	:	208-240 V~ 50/60 Hz
	Puissance	:	6/5 W
	Impédance Protected		
	Dimension	:	90 x 90 x 30 mm (environ)
	Adresse du fabricant	:	8 Rue Boutard 92200 NEUILLY FRANCE

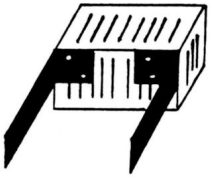
Pour qui est un rien bricoleur, le montage ne présentera aucune difficulté. Nous présentons ici deux possibilités concernant l'implantation du ventilateur selon l'existence dans le DAI d'un 'grand' radiateur ou l'inexistence de celui-ci.
Le but des explications qui suivent n'est pas de donner LA marche à suivre pour le montage mais UNE marche à suivre parmi d'autres, à vous de l'adapter avec les moyens du bord.

1er cas : le DAI est équipé d'un 'grand' radiateur
.....
Après avoir enlevé le 'capot' du DAI, démonter le couvercle protégeant l'alimentation, ensuite dévisser le radiateur (Attention aux composants qui y sont reliés).
Une fois détaché, scier le radiateur en deux parties égales (Atention à ne pas trop abimer le revetement noir).

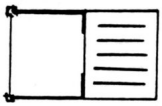


Jusqu'ici rien de bien sorcier mais les choses sérieuses ou plutot périlleuses vont commencer. Après avoir dévissé tous les composants attachés à la 'cage' de l'alimentation (coté carte vidéo), placer les deux radiateurs ainsi obtenus,à chaque extrémité de cette cage et percer très précautionneusement des trous (à travers le radiateur et la cage) afin de pouvoir les fixer solidement sur la cage.
REM vous pouvez déja utiliser les trous existants sur le radiateur .

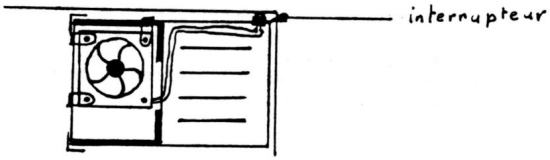




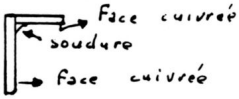
'Relier' les extrémités 'des' radiateurs afin de consolider le tout, à l'aide d'une 'barre' et de deux petites cornières (par exemple des pièces de mécanos) (voir dessin ci-dessous). Prendre garde à ne pas plier les radiateurs en forant les pour fixer les cornières.



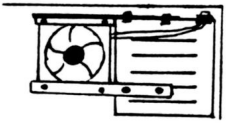
A ce stade, les opérations les plus délicates sont terminées, il reste à fixer le ventilateur sur l'armature ainsi construite à l'aide de petites cornières (voir dessin ci-dessous) et à brancher son alimentation sur l'interrupteur du DAI (si le ventilateur ne doit fonctionner que lorsque le DAI est allumé) ou sur un interrupteur extérieur (indépendant de celui d'allumage du DAI) .



2e cas : le DAI ne possède pas ce 'grand' radiateur
.....
Dans ce cas, il n'est plus possible d'utiliser le radiateur comme support, il faut trouver un substitut par exemple à partir d'une 'chute' d'époxy avec une face cuivrée (servant à la fabrication des circuits imprimés).
Scier la plaquette en trois partie : deux morceaux assez longs (environ 20 x 3 cm) et un morceau plus petit (environ 9.5 x 1.5).
Souder le petit morceau sur un grand afin de former un L.



Ensuite fixer cette plaque sur la 'cage' de l'alimentation (du coté des prises) à l'aide de vis et d'écrous (Les mêmes recommendations de prudence du premier cas pour le forage sont à observer).
Une fois fait, fixer la seconde plaquette, à la bonne distance, sur le couvercle de la 'cage' et forer les trous pour la fixation du ventilateur.



Pour terminer, fixer le ventilateur sur le support ainsi construit et brancher l'alimentation.

Et maintenant tous à vos outils,

Fabrice Duluins pour D.A.I.C. 7 Avril 1985

HARCOPY DU MODE 0 SUR IMPRIMANTE.

Ces deux petits programmes vont vous permettre d'imprimer soit une fenetre, soit l'intégralité d'un écran en MODE 0.

Vous avez certainement déjà tous un peu 'joué' avec des tableaux de chiffres (ou de texte) en MODE 0. Les données du tableau s'inscrivant petit-à-petit à l'aide de l'instruction CURSOR du BASIC. Si vous imprimez directement sur papier chaque donnée dans l'ordre d'arrivée à l'écran, vous aurez sur votre feuille une suite de chiffres qui n'aura plus rien à voir avec un tableau. Pour éviter cela, l'idéal serait donc de pouvoir copier toute la page texte sur imprimante quand toutes les données sont inscrites à l'écran. Il faudra donc réaliser une scrutation de l'écran !

Pour trouver l'adresse d'un caractère inscrit à l'écran, voici la formule à utiliser :

```
A$ = CHR$ ( PEEK ( #BFE7 - A * #86 - 2 * CURSX ) )
```

On met le caractère à trouver dans A$, l'adresse #BFE7 étant l'adresse la plus haute des caractères (CURSOR 0,23), A*#86 situe la position Y du point à trouver (A étant le nombre de ligne à soustraire à partir du haut de l'écran pour obtenir la position Y voulue et #86 étant l'espace entre chaque ligne.), 2*CURSX situe la position X du point recherché.

Voici maintenant deux application de cette formule :

Routine 1 : Une fois tout l'écran scruté et mis dans LIGNE$(x), la routine
..... imprime tout l'écran. (lignes 1120-1150)

Cette méthode n'est pas très rapide mais elle a certains avantages : en modifiant les coefficients de la boucle FOR/NEXT (ligne 1040), vous pouvez faire une copie d'une partie de l'écran, c'est-à-dire que vous n'etes pas obligé de copier tout l'écran quand vous n'avez besoin que d'une seule partie de celui-ci sur papier.

Routine 2 : La routine 2 est un peu plus astucieuse, mais comporte une
..... restriction : vous etes obligé de copier tout l'écran sur papier.

Cette routine scrute toujours la ligne supérieure de l'écran et puis l'imprime directement en-dessous de l'écran (et sur papier !). Avec le scrolling du DAI, tout l'écran remonte d'un cran et le programme rescrute la première ligne de l'écran (qui est en fait la deuxième de l'écran précédent) et l'imprime en dessous et ainsi de suite...

Cette seconde routine est beaucoup plus rapide que la première, du moins pour toutes les imprimantes qui ont un mini buffer d'une ligne comme, par exemple les EPSON, car on ne perd plus de temps à l'impression. L'imprimante ne 'ralentit' plus le DAI puisque son buffer d'une ligne imprime la dernière ligne de l'écran tandis que le DAI rescrute la première.

Les deux routines expliquées ici sont simples et faciles d'emploi, elles peuvent s'intercaler comme sous-routines dans un programme, elles ne sont pas longues, leur rapidité est acceptable, elles peuvent copier des fenetres et elles fonctionnent sur toutes les imprimantes (ce dernier point n'est pas à négliger car il existe d'autres routines d'hardcopy MODE 0 en langage machine mais ces dernieres ne 'tournent' que sur certains types d'imprimantes et ne sont pas plus rapide que la routine 2).

Ces routines peuvent certainement etre améliorées, si vous pouvez faire mieux, écrivez-nous, le sujet reste à approfondir !

Marc Vandermeersch

Pour D.A.I.C.

UTILITAIRE

```
10    REM ################################
11    REM ### HARD-COPY sur imprimante ###
12    REM ###                          ###
13    REM ###     sous-routine 1       ###
14    REM ###                          ###
15    REM ################################
16    REM ### Marc VANDERMEERSCH        ###
17    REM ###              pour DAIclic ###
18    REM ################################
120   REM *** Affichage de l'ecran a copier
130   LIST
140   GOSUB 1000
150   PRINT "Alors ??? ca marche non ???"
160   END
1000  REM *** sous-routine HARD-COPY
1010  CLEAR 2000:DIM TEXT$(24.0)
1020  ADR%=#BFE7
1030  ESP%=#86
1040  FOR CURSY%=23 TO 0 STEP -1
1050  LIGNE$=""
1060  FOR CURSX%=0 TO 59
1070  A%=23-CURSY%
1080  LIGNE$=LIGNE$+CHR$(PEEK(ADR%-A%*ESP%-2*CURSX%))
1090  NEXT CURSX%
1100  TEXT$(A%)=LIGNE$
1110  LIGNE$="":NEXT CURSY%
1120  PRINT CHR$(12):PRINT "Pressez 'SPACE' pour l'impression"
1130  POKE #131,0
1140  CALLM #D6DA:FOR T%=0 TO 23:PRINT TEXT$(T%):NEXT T%
1150  POKE #131,1:RETURN
```

```
10    REM ################################
11    REM ### HARD-COPY sur imprimante ###
12    REM ###                          ###
13    REM ###     sous-routine 2       ###
14    REM ###                          ###
15    REM ################################
16    REM ### Marc VANDERMEERSCH        ###
17    REM ###              pour DAIclic ###
18    REM ################################
110   REM *** Affichage
120   LIST
130   PRINT "Et maintenant, copions ! Quelle vitesse !!!"
140   GOSUB 1000
150   PRINT "Plus rapide, n'est-ce-pas ???"
160   END
1000  POKE #131,0
1010  ADR%=#BFE7
1020  FOR CURSY%=23 TO 0 STEP -1
1030  LIGNE$=""
1040  FOR CURSX%=0 TO 59
1050  LIGNE$=LIGNE$+CHR$(PEEK(ADR%-2*CURSX%))
1060  NEXT CURSX%
1070  PRINT LIGNE$
1080  NEXT CURSY%
1090  POKE #131,1:RETURN
```

MEMOIRE ECRAN

MODES 5, 6, 5A et 6A

1) Organisation de l'ecran

I) *** MODES 5 et 6 ***

L'ecran est divise en 256 lignes et 336 colonnes. Il y a toujours 2 octets qui contiennent l'information pour 8 points. Ceci entraine que pour une ligne on a besoin de (336/8)*2 = 84 octets afin de pouvoir representer les 336 points.

	0..7	8..15	...	328..335	
255	BFEB/BFEA	BFE9/BF0..7	8..15	...	328..335
255	BFEB/BFEA	BFE9/BF0..7	8..15	...	328..335
255	BFEB/BFEA	BFE9/BF0..7	8..15	...	328..335
255	BFEB/BFEA	BFE9/BF0..7	8..15	...	328..335
255	BFEB/BFEA	BFE9/BF0..7	8..15	...	328..335
255	BFEB/BFEA	BFE9/BF0..7	8..15	...	328..335
255	BFEB/BFEA	BFE9/BF0..7	8..15	...	328..335
255	BFEB/BFEA	BFE9/BF0..7	8..15	...	328..335
255	BFEB/BFEA	BFE9/BF0..7	8..15	...	328..335
255	BFEB/BFEA	BFE9/BFE8		BF99/BF98	
254	BF91/BF90	BF8F/BF8E		BF3F/BF3E	
253	BF37/BF36	BF35/BF34		BEE5/BEE4	
.					
.					
.					
2	66F9/66F8	66F7/66F6		66A7/66A6	
1	669F/669E	669D/669C		664D/664C	
0	6645/6644	6643/6642		65F3/65F2	

II) *** MODES 5A et 6A ***

L'ecran ne donne plus que les lignes 0 a 211.

	0..7	8..15	...	328..335
211	BFEB/BFEA	BFE9/BFE8		BF99/BF98
210	BF91/BF90	BF8F/BF8E		BF3F/BF3E
.				
.				
.				
2	7671/7670	766F/766E		761F/761E
1	7617/7616	7615/7614		76C5/76C4
0	75BD/75BC	75BB/75BA		756B/756A

III) *** MODES 5, 6 / 5A, 6A ***

A gauche et a droite de ces octets se trouvent encore deux autres octets (ce qui fait 8 points) qui ne sont pas accessibles par le basic. La ligne 0 dans le MODE 5 ou 6 est alors la suivante :

- 2 octets de controle :6649 , 6648
- 2 octets pour le bord :6647 , 6646
- 84 octets pour la ligne :6645 -> 65F2
- 2 octets pour le bord :65F1 , 65F

L'ecart entre deux lignes est de 90 decimal.
L'adresse a laquelle est code un point de coordonnees (x,y) peut alors etre determinee par la formule suivante :

```
adr:=#6645-INT(x/8)*2 + 90*y
```

Le point en question est alors code en adr et adr-1.

2) COMPOSITION DES DEUX OCTETS

I) *** MODE 6, 6A ***

La composition des octets sera expliquee a l'aide d'un exemple.
Analyse des deux octets #6645 et #6644 qui contiennent les points 0 a 7 de la ligne 0 dans le MODE 6.
Apres execution des instructions :

```
COLORG 8 0 15 4
DOT 0,0 0
DOT 3,0 15
DOT 6,0 4
```

l'occupation des octets est la suivante :

```
                          7 6 5 4 3 2 1 0 - valeur octet
                          0 1 2 3 4 5 6 7 - position du dot
adr.:#6644 := #82 = 1 0 0 0 0 0 1 0
adr.:#6645 := #12 = 0 0 0 1 0 0 1 0
                          0 8 8 F 8 8 4 8 - information de couleur
                                            (F correspond a 15)
```

Un bit qui vaut '1' dans un des deux octets entraine que la position du dot correspondant a ce bit est mis en couleur. La couleur de ce point est fonction de l'octet dans lequel le bit a ete mis a '1'.
Si l'instrction est COLORG C1 C2 C3 C4 on retrouve la couleur:

C1 si les 2 bits correspondant au dot en question valent '0' (couleur de fond).

C2 si le bit dans l'adresse basse est mis a '1' et que le bit de l'adresse haute vaut '0' (bit 7 de #6644 vaut '1' et bit 7 de #6645 vaut '0' => couleur 0 en 0,0).

C3 si le bit dans l'adresse basse vaut '0' et que le bit de l'adresse haute vaut '1' (bit 4 des deux octets pour le dot 3,0).

C4 si les deux bits valent '1' (bit 1 pour le dot 6,0).

II) *** MODE 5,5A ***

Comme exemple on va choisir en mode 5A, les points 0 a 7 de la ligne 0. Dans le mode 4 couleurs les deux octets contiennent l'information concernant la couleur et la position. Dans un mode 16 couleurs, chaque octet joue son role. L'octet d'adresse basse contient l'information concernant la couleur tandis que l'octet d'adresse haute contient uniquement l'information concernant la position.

L'exemple suivant :

```
COLORG 8 0 0 0
MODE 5A
DOT 0,0 0
DOT 3,0 0
DOT 6,0 0
```

choisit comme couleur de fond le gris (8). Tous les points dessines apparaissent en noir (0). L'occupation de #75BC et #75BD est alors la suivante:

```
                          7 6 5 4 3 2 1 0 - valeur du bit
adr.:#75BC : = #08 =      0       8
adr.:#75BD : = #92 = 1 0 0 1 0 0 1 0
                          0 1 2 3 4 5 6 7 - valeur du dot
```

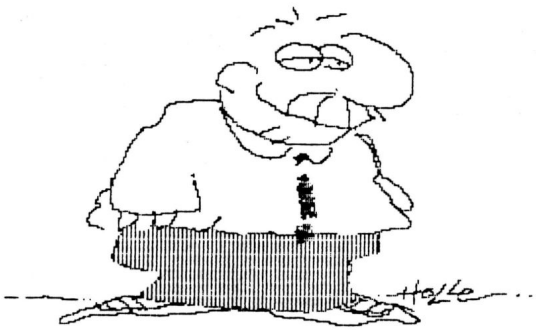
Tous les bits mis a '1' de l'octet d'adresse elevee determinent comme couleur celle qui est fixee par les 4 bits de poids plus fort de l'octet d'adresse basse. La couleur qui represente les '0' de l'octet d'adresse plus elevee est celle qu'on trouve dans les 4 bits de poids plus faible de l'octet d'adresse basse.
De cette reflexion, on peut facilement deduire que l'on peut au maximum

retrouver 2 couleurs pour representer les 8 points. Il reste a remarquer que l'occupation des deux octets aurait pu etre la suivante:

```
adr.: #75BC := #B0 =     8       0
adr.: #75BD := #6D = 0 1 1 0 1 1 0 1
```

Il est facile de voir que l'effet aurait ete le **meme**.

Texte original en allemand: Stefan GOLLER.
Traduction: Andres KURT (Octobre 84),
BUTGENBACH (B).



ASSEMBLER

==========
== SAUVEGARDE DE L'ECRAN GRAPHIQUE ==
==========

Vous avez certainement déjà désiré avoir la possibilité de sauver momentanément un dessin de l'écran graphique pour différentes raisons.

Soit parceque vous désiriez changer de mode notamment pour éditer une partie d'un programme, soit que vous vouliez modifier une partie du dessin avec la possibilité de le récupérer sans l'avoir détruit, soit encore que vous envisagiez carrément un programme d'animation.

Il y a plusieurs solutions à ce problème, mais toutes n'ont pas le meme domaine d'application.

1. Solution évidente: Sauver la zone d'écran en question sur cassette à l'aide de la commande WRITE. Ceci offre la possibilité d'un stockage quasi illimité (suivant le budget cassettes...) mais c'est très lent!

2. Solution proposée: Il s'agit de la routine ci-dessous.

Cette routine recopie le contenu d'une zone mémoire vers une autre et peut d'ailleurs servir à tout autre chose que des sauvetages graphiques. La rapidité de cette méthode est telle qu'une certaine forme d'animation peut etre envisagée.

Supposons que nous voulions déplacer le contenu de la zone A vers la zone B. Le principe est de:

- Placer l'adresse de début de la zone A en #303/#302.
- Placer l'adresse de début de la zone B en #301/#300.
- Placer la taille de A en #305/#304.
- Exécuter la routine par CALLM #320.

Exemple d'utilisation:

Le petit programme BASIC ci-joint permet de sauver et de récupérer un graphique réalisé en MODE 4.

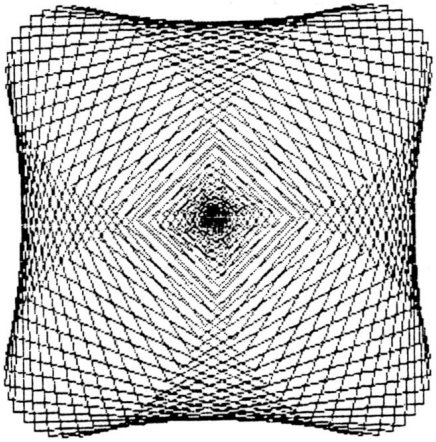
- L'adresse la plus basse de l'écran en MODE 4 est #A894.
- La taille de la mémoire écran en MODE 4 est: #BFEF-#A894+1=#175C.

```
]#L.
PAGE 01

001                          ORG   :320
002 0320 C5                  PUSH  B
003 0321 D5                  PUSH  D
004 0322 E5                  PUSH  H
005 0323 F5                  PUSH  PSW
006 0324 F3                  DI
007 0325 2A0003              LHLD  :300
008 0328 44                  MOV   B,H
009 0329 4D                  MOV   C,L
010 032A 2A0203              LHLD  :302
011 032D 54                  MOV   D,H
012 032E 5D                  MOV   E,L
013 032F 2A0403              LHLD  :304
014 0332 1A                  LDAX  D
015 0333 02                  STAX  B
016 0334 03                  INX   B
017 0335 13                  INX   D
018 0336 2B                  DCX   H
019 0337 E600                ANI   0
020 0339 B4                  ORA   H
021 033A B5                  ORA   L
022 033B C23203              JNZ   :332
023 033E FB                  EI
024 033F F1                  POP   PSW
025 0340 E1                  POP   H
026 0341 D1                  POP   D
027 0342 C1                  POP   B
028 0343 C9                  RET
029 0344                     END


>D320 343
0320 C5 D5 E5 F5 F3 2A 00 03 44 4D 2A 02 03 54 5D 2A
0330 04 03 1A 02 03 13 2B E6 00 B4 B5 C2 32 03 FB F1
0340 E1 D1 C1 C9
```

```
60000 PRINT CHR$(12)
60010 PRINT "J'espere que vous avez bien charge le programme"
60020 PRINT "ROUTINE DE DEPLACEMENT D'OCTETS!"
60030 PRINT :PRINT "Sauvetage ou Recuperation (S/R)?";
60040 A%=GETC:IF A%=0 THEN 60040
60050 IF A%=ASC("S") THEN A1%=#300:A2%=#302:GOTO 60080
60060 IF A%=ASC("R") THEN A1%=#302:A2%=#300:GOTO 60080
60070 GOTO 60040
60080 PRINT CHR$(A%)
60090 PRINT :INPUT "Adresse du transfert";ADR%:PRINT
60100 ADR1%=INT(ADR%/256.0)
60110 ADR2%=ADR%-ADR1%*256
60120 POKE A1%,ADR2%:POKE A1%+1,ADR1%
60130 POKE A2%,#94:POKE A2%+1,#A8
60140 POKE #304,#5C:POKE #305,#17
60150 MODE 4
60160 CALLM #320
60170 END
```

GRAPHIQUE

```
1      REM QUADERS
10     MODE 5
20     FILL 0,0 XMAX,YMAX 0
40     Q%=RND(15.0)+1.0
41     C%=RND(80.0)+15.0
42     S1%=SQR(C%^2.0/2.0)
43     S2%=S1%
47     X%=RND(XMAX-S1%-70.0)
48     Y%=RND(YMAX-S2%-70.0)
49     B%=RND(XMAX-S1%-X%-15.0)+15.0
50     A%=RND(YMAX-S2%-Y%-15.0)+15.0
100    IF GETC<>0.0 THEN 10
200    GOSUB 10000
300    GOTO 40
10000  REM
10100  DRAW X%,Y% X%+B%,Y% Q%
10120  DRAW X%+B%,Y% X%+B%,Y%+A% Q%
10140  DRAW X%+B%,Y%+A% X%,Y%+A% Q%
10160  DRAW X%,Y%+A% X%,Y% Q%
10300  DRAW X%,Y%+A% X%+S1%,Y%+A%+S2% Q%
10320  DRAW X%+B%,Y%+A% X%+S1%+B%,Y%+A%+S2% Q%
10340  DRAW X%+B%,Y% X%+S1%+B%,Y%+S2% Q%
10400  DRAW X%+S1%,Y%+A%+S2% X%+S1%+B%,Y%+A%+S2% Q%
10420  DRAW X%+S1%+B%,Y%+S2% X%+S1%+B%,Y%+A%+S2% Q%
10500  FOR I%=X%+S1% TO X%+S1%+B% STEP 4
10520  DOT I%,Y%+S2% Q%
10540  NEXT
10560  FOR I%=Y%+S2% TO Y%+S2%+A% STEP 4
10580  DOT X%+S1%,I% Q%
10600  NEXT
10620  FOR I%=X% TO X%+S1% STEP 3
10640  DOT I%,(S2%/S1%)*(I%-X%)+Y% Q%
10660  NEXT
10900  RETURN
```

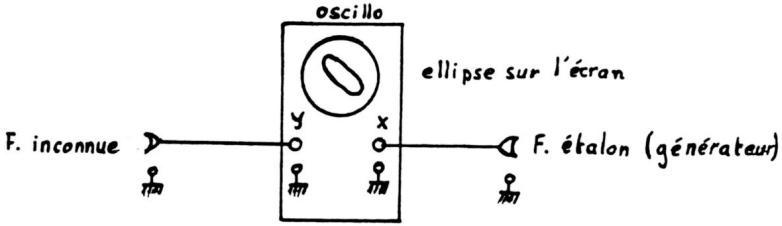


L'ORDINATEUR,ETALON DE FREQUENCE

Prenons un cas de figure somme toute banal:mordu de l'informatique, passionné d'électronique,vous venez d'achever la construction d'un circuit qui doit osciller sur des fréquences précises et qu'il ne "reste" qu'à règler.Bien entendu,amateur chevronné, vous disposez-outre du fer à souder-d'un générateur,d'un oscilloscope,d'un fréquencemètre (mais de performances modestes, bien entendu !).
Résultat:un réglage médiocre, à moins que vous ne vous rappeliez à temps que votre ordinateur peut vous etre d'un précieux secours !

I.LE CALVAIRE DE L'ELECTRONICIEN PEU FORTUNE.

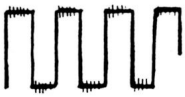
La "bete" à régler est un MODEM (1) aux normes françaises.Il lui faut donc générer deux couples de fréquences sinusoidales 980-1180 Hz et 1650-1850 Hz signalant respectivement les 1 et 0 logiques des modes appel et réponse.Rien à voir avec les U.H.F. (2) :ce sont des basses fréquences aisément manipulables (meme un magnétocassette les restitue sans peine,une ligne téléphonique à fortiori !).OUI,MAIS il faut se caler au Herz près (il vaudrait mieux,en fait !) et c'est là que les ennuis commencent:l'oscilloscope,meme avec base de temps étalonnée (3),graticule (4) gradué en unités de temps,est incapable de faire sérieusement la différence entre 980 et 1000 Hz par exemple.C'est là bien sur que le fréquencemetre intervient. OUI,MAIS ces charmants appareils,souvent prévus pour lire des fréquences astronomiquement hautes (des centaines de mégaherz) sont lamentables du coté des basses valeurs.Le mien,dont je tairai pudiquement le nom,a une résolution MINIMALE de 100 Hz (et pas de fonction périodemètre (5),ce qui aurait pu améliorer les choses).Reste à procéder par comparaison de fréquences, c'est-à-dire observer sur l'oscilloscope ce qu'il est convenu d'appeler une figure de LISSAJOU (6).La fréquence-étalon passe par l'entrée horizontale de l'oscilloscope,la fréquence à comparer par l'entrée verticale. L'égalité se traduit théoriquement par une ligne orientée à 45 degrés et en réalité par un cercle ou une ellipse (à cause des déphasages).Si l'ellipse est rigoureusement stable,les deux fréquences sont rigoureusement identiques,sinon l'ellipse tourne autour d'un axe vertical à une vitesse qui est fonction de l'écart de fréquence (un tour par seconde correspond en gros à un écart de 1 Hz).Cela suppose une fréquence-étalon parfaitement stable et de hauteur exactement connue,toutes choses qu'un générateur bon marché est incapable de fournir,hélas !



Obtention d'une figure de Lissajou.

II.LE SECOURS DE L'ORDINATEUR.

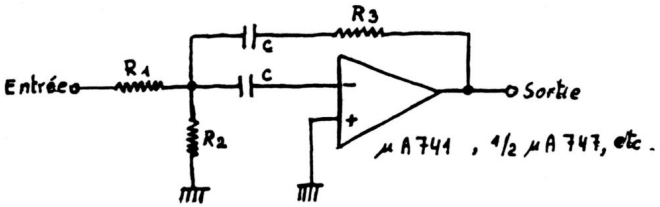
Beaucoup d'ordinateurs domestiques sont maintenant pourvus d'un ou plusieurs générateurs sonores.Le DAI par exemple peut émettre des fréquences allant de 31 Hz à 1 MHz (en principe),avec une stabilité qui est celle du quartz de son horloge.Point précieux à noter:l'interpréteur basic demande directement la fréquence en HERZ (pas de POKE's compliqués) dans une variable qui peut etre virgule flottante.Pour notre réglage,et avec les six chiffres significatifs du DAI,il est donc possible despérer une précision de l'ordre du centième de Hz,ce qui est une belle performance !
Un écueil de taille cependant:ces fréquences sortent en signaux rectangulaires affectés de pics (résidus d'échantillonnage) et sont rigoureusement impropres à l'obtention d'une figure de Lissajou lisible.



Signaux à la sortie "Stéréo" du DAI

Donc,il faut les convertir en sinusoides, ce qui s'obtient par suppression des harmoniques (filtre passe-bande Butterworth du 2ème ordre.Ce nom est prétentieux,mais c'est simple à bricoler !)

Les amplis opérationnels sont là pour nous sauver (741,747,etc)



GAIN: $R_3/2R_1$
BANDE PASSANTE: $2/R_3.C$
FREQUENCE CENTRALE: $\sqrt{(1/R_3.C^2(1/R_1+1/R_2))}/2\pi$
(R en Ohms,C en Farads,F en Herz)
Prendre R1=10*R3 (gain de 5) et R2=510 Ohms pour le calcul et mettre à la place une résistance ajustable de 1Kohms.
Pour ma part,j'ai utilisé pour le couple 980-1180 Hz (Fo=1080)
R1=4,7k ;R2=510 Ohms;R3=47k;C=33nF
et pour 1650-1850 Hz (Fo=1750)
R1=12k;R2=510 Ohms;R3=120k;C=12nF
et j'ai monté chaque fois trois cellules de filtre en série, afin d'avoir un gain de 125 (le DAI "sort" environ 400 mV crete,et l'entrée horizontale de mon oscilloscope n'est pas très sensible).
Tout ceci se bricole en quelques minutes sur une plaque d'expérimentation,et l'alimentation (+ et - 12 V) était prise directement sur le modem.
Le réglage,pour une fréquence considérée,est ultra-simple: injecter à l'entrée le signal à la fréquence voulue venant du DAI grace au "programme" suivant:

```
SOUND 0 0 15 0 FREQ(980):REM PAR EXEMPLE
```

Régler les ajustables de 1kOhms pour un signal maximum en sortie de filtre.On obtient une sinusoide parfaite d'environ 15 v crete,ce qui est plus que suffisant.

III.UN REGLAGE EPOUSTOUFLANT

Restait d'abord à apprécier la qualité du calage initial du modem au fréquencemètre.Donc,sortie modem sur l'entrée verticale de l'oscilloscope,sortie DAI via les filtres sur l'entrée horizontale,et programme suivant:

```
10 ENVELOPE 0 16:REM ENVELOPPE No 0 AVEC VOLUME MAXIMUM
20 F=900
30 SOUND 0 0 15 0 FREQ(F):REM ENVELOPPE 0,CANAL 0,SON NORMAL
40 PRINT F
50 F=F+1
60 GOTO 30
```

Il faut bien sur arreter le déroulement du programme <BREAK> dès qu'apparait l'ellipse sur l'oscilloscope (attention:cela va vite !).
Résultat:calage sur 937 Hz au lieu des 980 prévus.Hum !
Nouveau programme pour le calage exact de cette fréquence:

```
10 ENVELOPE 0 16
20 INPUT"FREQUENCE:";F:PRINT
30 SOUND 0 0 15 0 FREQ(F)
40 GOTO 20:REM ATTENTE NOUVEL INPUT,MAIS N'ARRETE PAS LE SON
```

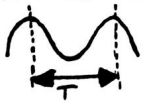
A ce stade,si le réglage ne peut immobiliser complètement l'ellipse,on peut APPRECIER l'écart à l'accord parfait en entrant des valeurs décimales puis centésimales (980.1 puis 980.2 ou encore 980.11 puis 980.12 etc.).Le sens et la vitesse de révolution de l'ellipse renseignent sur la nature de l'écart. On peut considérer qu'une révolution toutes les minutes constitue un assez joli résultat (et que l'oscillateur du modem jouit d'une belle stabilité !)
Il ne restait plus qu'à répéter 4 fois ces opérations pour avoir les 4 fréquences calées avec une précision de l'ordre du demi-Herz (modem fonctionnant depuis une heure)

Pour la petite histoire,j'ai construit également un autre modem,aux normes américaines,avec un circuit intégré spécialisé, prévu pour donner directement des fréquences exactes,grace à une horloge interne à quartz.La tentation était grande de

vérifier son exactitude.Eh bien,je suis très content de mes deux modems !

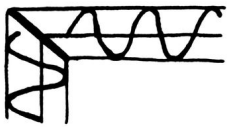
ALAIN MARIATTE.

(1) MODEM:appareil utilisé pour connecter deux ordinateurs par le réseau téléphonique.
(2)U.H.F.:ultra high frequencies,de l'ordre de centaines de MHz;télévision,par exemple.
(3) BASE DE TEMPS:oscillateur réglable,gradué en temps par division d'écran (1 micro seconde /div. par exemple) qui commande le balayage de gauche à droite du spot sur l'écran de l'oscilloscope.Ces graduations permettent de calculer la fréquence du signal observé (F= 1 / T secondes).
(4) GRATICULE:graduations (divisions) de l'écran de l'oscillo.
(5) PERIODEMETRE:le fréquencemètre donne la période du signal et non plus sa fréquence,c'est-à-dire la durée (ou distance) qui sépare deux sommets consécutifs d'une sinusoide,par ex.



Cela permet une lecture plus précise des basses fréquences.

(6) LISSAJOU:a montré que la composition orthogonale de deux sinusoides formait un segment de droite incliné à 45 degrés.



Le Club I.D.C. Bordeaux vous presente un montage destine a soulager vos maigres Paddles :

Interfacage DAI avec des joysticks type ATARI ou spectravideo

Ces joysticks ont une difference fondamentale avec des paddles (Pdl), les directions opposees (Droite-Gauche Haut-Bas) agissent sur des interrupteurs et non sur des potentiometres.

Mais peut etre est il bon de rappeler le fonctionnement de base du Pdl. Son mouvement fait varier une resistance (potentiometre) situee electriquement entre le +5 Vcc et l'entree Pdl consideree.
Cette variation va de 0 ohms a 100 Kilo ohms faisant prendre a la variable Pdl des valeurs de 0 a 255.

Il faudra donc simuler , pour que le montage reste compatible avec les programmes existants , un comportement identique a celui d'un Pdl n'ayant que trois positions (pour chaque dimension).

D'ailleurs, pour simplifier les schemas, nous ne considererons qu'une seule dimension : celle des X.

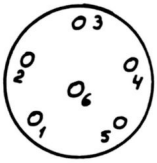
Un petit tableau pour garder claires a l'esprit les relations entre la position physique du manche et la valeur associee de la variable Pdl ainsi que de la resistance :

Position du manche	=	A GAUCHE	AU MILIEU	A DROITE
Valeur de Pdl	=	0	125	255
Valeur de la resistance	=	0 ohm	50 Kohms	100 Kohms

REALISATIONS HARDWARE

Schéma des prises PDL et Joystick

Entrée DAI



1 = +5 Vcc

2 = entrée PDL y

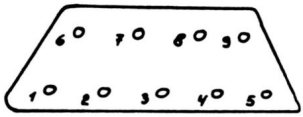
3 = entrée PDL z

4 = bonton (event)

5 = entrée PDL x

6 = masse (GND)

Sortie Joystick



1 = Haut

2 = Bas

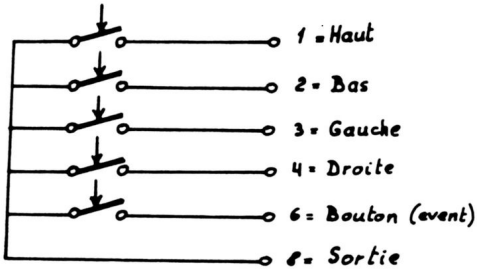
3 = Gauche

4 = Droite

6 = bouton

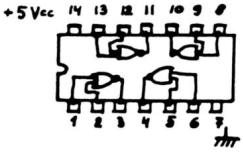
8 = sortie

Equivalent électrique de l'intérieur du joystick.



Brochage du 4011

(vue du dessus)



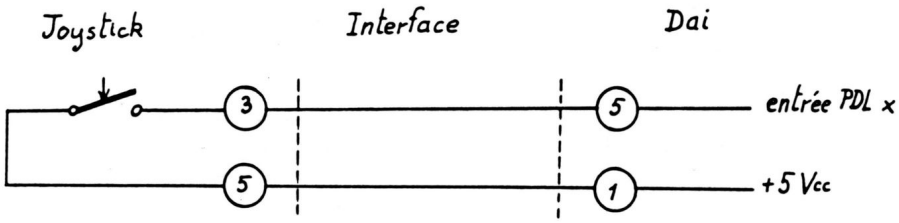
Le C.I. CD4011 comporte 4 portes "Non-Et" (NAND) à deux entrées.

Pour avoir la valeur 0 , il faut intercaler , entre l'entree Pdl et le + 5 Vcc de la prise Pdl , une resistance nulle.

Aucun probleme pour cela avec l'interupteur actionne par le joystick, quand on l'incline a gauche :
il suffirait d'intercaler cet interupteur entre notre + 5 Vcc et l'entree Pdl du DAI

Il apparait donc que , pour faire prendre a l'entree Pdl la valeur 0 i.e. a fond vers la gauche il suffirait de :

"Interface" 1

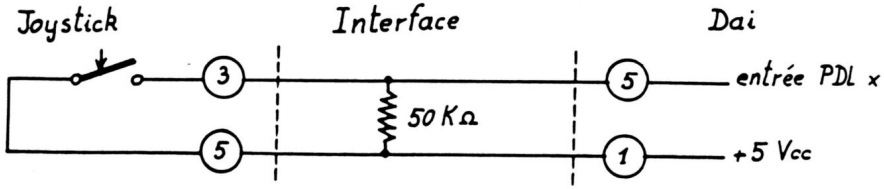


Mais cette solution ne serait valable que lors du maintient de la manette a gauche , car pour les autres positions , le circuit electrique est ouvert et notre variable Pdl prend n'importe quelle valeur.

Il faut donc que , lorsque le manche reste au milieu , une valeur approximative de 125 soit prise par la variable Pdl , ce qui correspond a l'introduction d'une resistance d'environ 50 Kilo ohms entre le + 5 Vcc et l'entree Pdl

On est donc contraint de modifier notre premiere ' interface ' :

"Interface" 2



Le resultat escompte est bien atteint :

--> lorsque le manche est au milieu , la resistance entre l'entree Pdl et le + 5 Vcc du DAI , de 50 Kilo ohms , amene bien la variable Pdl a 125 et :

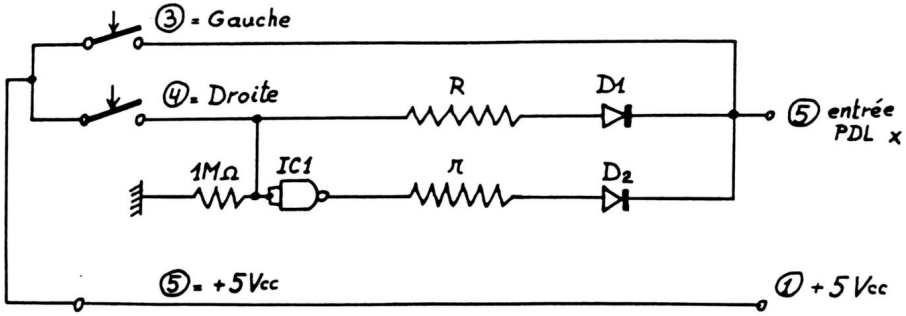
--> lorsque le manche est vers la gauche :
la resistance est court-circuitee et l'entree Pdl prend la valeur 0

MAIS HELAS , tout se complique si l'on veut , en plus , pouvoir aller vers la droite (y en a quand meme qui ne sont jamais satisfaits !!)

Car pour avoir une valeur de 255 pour notre entree Pdl , il faudrait que la resistance de 50 Kilo ohms se transforme en 100 Kilo ohms lors de l'action de l'interrupteur du Joystick 'vers la droite' tout en respectant les 50 Kilo ohms lorsque le manche est au milieu et donc aucun interrupteur actionne.

Ce serait chose facile avec des inverseurs au lieu d'interrupteurs mais comme il n'en est rien , il va falloir en simuler un avec une porte logique , la plus connue , valant trois francs (francais !) six sous (c'est pas le PEROU ...) , s'alimentant grace au + 5 Vcc de cette meme entree Pdl (ca consomme si peu) , j'ai nomme le ' CD-4011 '

"Schéma pour une dimension"



- Il decoule de ce montage que :

--> Lorsque ' AUCUN ' interrupteur n'est actionne :

* La resistance de 1 Mohm maintient un niveau logique 0 sur les deux entrees reunies de notre porte NON ET (NAND pour les Anglophiles).

* Donc sa sortie est a un niveau logique 1 (+ 5 Vcc) qui est transmis a l'entree Pdl , via la resistance 'r' et la diode D2 (dont le role sera explique plus loin).

* La valeur theorique de cette resistance 'r' est de 50 Kohms pour une valeur de la variable Pdl de 125.

--> Lorsque l'interrupteur ' GAUCHE ' est actionne :

* Comme precedement , la mise en parallele avec 'r' d'une resistance nulle , amene la valeur de la variable Pdl a 0.

--> Lorsque l'interrupteur ' DROITE ' est actionne :

* L'entree de la porte NON ET est portee au + 5 Vcc = 1 logique , sa sortie passe donc a 0 Vcc = 0 logique, donc aucun courant ne passe plus dans 'r'.

* Par contre le + 5 Vcc est transmis a l'entree Pdl , via la resistance 'R' et la diode D1.

* La valeur theorique de cette resistance 'R' est de 100 Kohms pour une valeur Pdl de 255.

- Le ROLE de D1 et D2

--> D1 sert a empecher le 1 logique , present a la sortie de la porte NON ET , lorsque aucun interrupteur n'est actionne , de venir se reboucler sur l'entree de cette meme porte , via 'r'-D2->D1<-'R' , et ainsi faire basculer sa sortie a 0.

--> D2 sert a empecher le + 5 Vcc , present lors de l'action de l'interrupteur ' DROITE ' , de venir se court-circuiter avec le 0 Vcc = 0 logique alors present en sortie de la porte NON ET via 'R'-D1->D2<-'r' , et ainsi faire fumer notre pauvre porte ...

*** J'ai jusqu'a present parle de valeurs theoriques pour 'r'. En effet , si son calcul est aise lorsqu'on veut obtenir une valeur dl de

125 --> 'r' = 50 Kohms, il en va tout autrement dans la realite lors ue notre Joystick doit s'adapter a des jeux differents dont les bornes de changement de direction ne sont pas les memes ....

... ou sont parfois farfelues!

Il nous faut donc passer de la notion de valeur ponctuelle a celle de plage d'action :

Un EXEMPLE :

```
-----------------------------------------------------------------------------

                          GAUCHE            NE  BOUGE  PAS          DROITE
Direction du mobile }----------------{}-----------------{}---------------{

Valeur du Pdl     :  0-------------100??------125------150??------------255

Valeur Joystick   :  0  -    -    -    * de 100 a 150 *   -    -    -   255

-----------------------------------------------------------------------------
```

Si pour un certain jeu , le mobile est fixe pour une valeur de Pdl comprise entre 100 et 150 , pour le jeu suivant , il se peut que le concepteur ait prefere des bornes de 150 a 200 (ca se voit !!).

Il en decoule immediatement que , pour le calcul du 'r' de notre montage si 'r' = 50 Kilo ohms --> Pdl = 125 : notre mobile ira toujours vers la Gauche alors que le Joystick est au repos.

Notre mobile ne sera jamais immobile ... (sic !).

Pour qu'il le soit il faudrait 'r' aux alentours des 60 Kilo ohms.

Avec un 'r' fixe c'est impossible

Pour parer aux differents cas de figures rencontrables il suffit d'intercaler en serie :

* une resistance ajustable de 33 Kilo ohms

* une resistance fixe de 33 Kilo ohms

```
                33 K ohms    33 K ohms

            o---||||||---o---||||||---o
                        I      @
                        I------I


              'r'FIXE   , r' AJUSTABLE

            }--------------------------{
                      'r' Total
```

Ce qui permet de faire varier 'r' de 33 Kilo a 66 Kilo ohms.

- La dimension fort reduite de l'ensemble des elements autorise leur montage a l'interieur du Joystick , en menageant un petit trou de 3 mm de diametre en face de chaque resistance ajustable (une pour DROITE-GAUCHE, une autre pour HAUT-BAS) afin d'en faciliter le reglage.

Il ne reste plus qu'a remplacer la prise sur le fil du Joystick par une fiche DIN , conformement aux plans joints, seuls 5 des 6 fils seront utilises : le "5" au + 5 Vcc , le "3" pour les 'X' , le "1" pour les 'Y', le "6" pour le 'Bouton' et , soit le "2" soit le "4" , pour la 'Masse'.

Cependant , cette modification ne permet plus l'utilisation de votre Joystick sur d'autres machines !!

Aussi pour ceux qui empruntent un manche a un ami , il faudra faire les frais d'une prise 'Male 9 broches' sur laquelle se branchera le Joystick , afin de realiser l'interface dans une petite boite.

Ainsi le cablage du Joystick ne sera pas touche :

* Il faudra amener 5 fils , du DAI au boitier interface.

* La 'Masse' ne servant qu'aux connections du CD-4011 , elle ne sera pas cablee sur le connecteur du Joystick.

* Par contre les 6 fils du Joystick seront exploites conformement au schema de fin d'article.

Cette derniere solution , un peu plus onereuse , est la plus souple et ne greve en rien l'avenir de vos Joysticks.

Idee de l'Interface I.D.C. Bordeaux

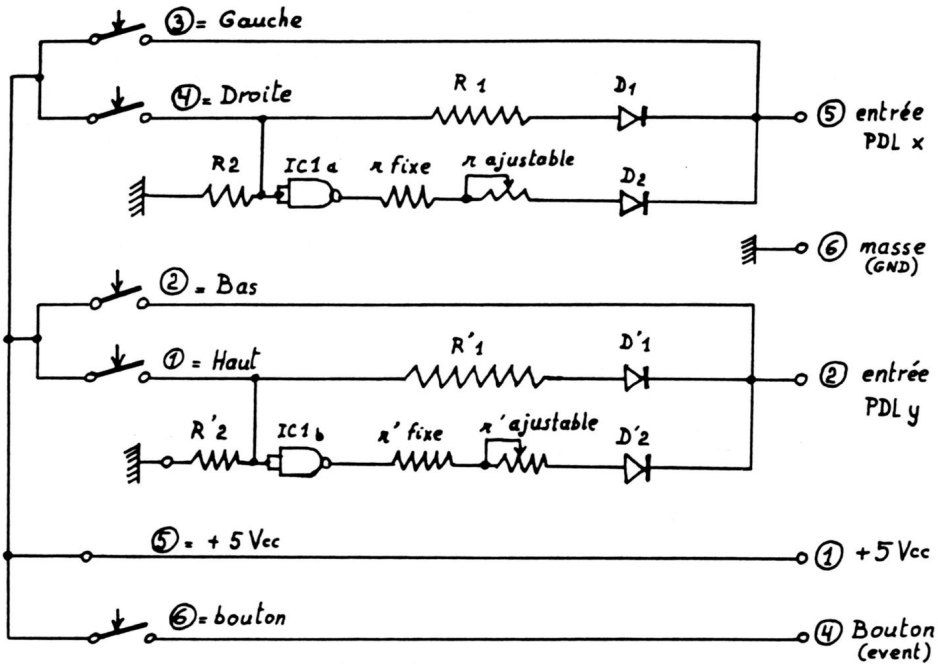
Joystick avec sa prise 9 broches	{ Prise idem { au Joystick	* montage * I.D.C.Bx.	DAI avec sa prise 6 broches

J'espere que cette petite etude realisee par l'atelier du club I.D.C. Bordeaux vous aura eclaire sur le sujet et je demande pardon aux ' Afficionados ' du fer a souder pour toutes les redondances a l'intention des neophites.

LE PRESIDENT : Delannay Bruno.

Schéma complet

pour un Joystick à 2 axes : x = Gauche - Droite
y = Bas - Haut



Nomenclature des composants.

- R1 = R'1 = 100 KΩ
- R2 = R'2 = 1 MΩ
- r fixe = r' fixe = 33 KΩ
- r ajustable = r' ajustable = 33KΩ ajustable.
- D1 = D'1 = D2 = D'2 = ce qui tombe sous la main, par exemple la 1N914
- IC1 = 4011 (seulement 2 portes sur les 4 sont utilisées)

AMD 9511

LE PROCESSEUR ARITHMETIQUE AMD 9511

Nous savons, grace au manuel, que le DAI peut etre équipé d'un 9511. Mais nous restons sur notre faim pour ce qui est de ses caractéristiques, de son emploi,... Le 9511 est un circuit intégré de deux fois douze broches. Il se loge à l'intérieur du DAI sur le support libre, à coté des ROM, son repère dirigé vers le connecteur X-BUS, comme le sont les ROM.

Le manuel français (page 141) nous indique comment repérer la présence de ce circuit par un PRINT PEEK(#FB00). S'il est présent, la réponse sera: DIVISION BY ZERO. Mais la lecture en #FB00 est interdite (par HARD) et rendra notre PC bien malade, de telle façon qu'une action sur RESET est nécessaire pour tout remettre en ordre. Un autre point à signaler: certains de vos programmes en LM risqueront de ne plus fonctionner correctement (routines PADDLES, MATH,...). Faut-il donc modifier tous ces programmes ou sortir le processeur? Et bien heureusement non. Il suffit de jouer sur l'indicateur #D4.

```
POKE #D4,#00        "déconnecte" le processeur arithmétique
POKE #D4,#7B        le met en service
```

Attention, ne mettez pas d'autres valeurs que celles indiquées au risque de devoir se servir du RESET.

Pour vous donner un idée, ce petit test:

```
IMP FPT
10 FOR A=0 TO 10000
20 X=(voir tableau)
30 NEXT A
```

FONCTIONS	LIGNE 20	TEMPS D'EXEC. AVEC 9511	(MIN:SEC:CEN) SANS 9511
boucle vide		00:07:04	00:06:99
ABS	X=ABS(-253.7)	00:33:16	00:35:12
ACOS	X=ACOS(0)	00:54:46	03:18:11
ALOG	X=ALOG(0.3020)	00:48:26	08:19:70
ASIN	X=ASIN(1)	00:27:53	00:31:80
ATN	X=ATN(A)	00:52:52	06:01:58
COS	X=COS(A)	00:47:78	07:15:58
EXP	X=EXP(-1)	00:51:53	07:32:82
LOG	X=LOG(A+1)	00:55:58	06:40:32
LOGT	X=LOGT(A+1)	00:56:42	07:10:50
SIN	X=SIN(A)	00:45:90	07:07:64
SQR	X=SQR(A)	00:28:06	02:49:59
TAN	X=TAN(A)	00:51:71	14:56:52
↑	X=A↑3	01:15:40	13:24:80
*	X=A*A	00:31:38	00:49:51
/	X=1/(A+1)	00:49:75	01:36:73

Fabien FOJUD
F-57 CREUTZWALD.

Dossier ASSEMBLEUR

```
-------------------------------------------------------------------------
I                                                                       I
I              M I C R O P R O C E S S E U R      PART 1                I
-------------------------------------------------------------------------
```


JANVIER 85

D.A.I.C.L.I.C

par HP LEGRY
F-59 DOUAI

LE BINAIRE ET L'HEXADECIMAL

Peut-etre etes vous desireux d'approcher ce mystere que represente la programmation en assembleur ou langage machine. Il vous faudra alors utiliser avec beaucoup de dexterite les nombres binaires et les nombres hexadecimaux.

Cet article vous aidera a comprendre et utiliser ces nombres. Ne negligez pas ces lignes si vous n'y connaissez rien. On ne peut pas programer en assembleur sans connaitre ces bases fondamentales.

1 - DEFINITIONS

A - LE BIT

Le bit est un nombre qui ne prend que deux valeurs (0 ou 1). Vous en rencontrez tous les jours : l'etat d'une lampe (allume ou eteinte), ou celui d'une porte (ouverte ou fermee).

Il est utilise pour caracteriser une variable qui ne prend que deux etats. Si j'associe la variable L a une lampe par exemple :

L=1 si la lampe est allumee
L=0 si la lampe est eteinte

L est alors codee sur un bit

B - LA BASE 2

On utilise couramment en arithmetique la base 10 (decimale). Dix symboles sont necessaires pour definir ou coder un chiffre (0 a 9). Lorsque les dix symboles sont utilises, on "retient" une base et on recomence au premier synbole.

* Exemple :

chiffre 5 : 5
chiffre 15 : une base (10) + 5 ,soit 1 X (10) + 5
par definition, on accole les deux chiffres : 1 5
Le phenomene se reproduit lorsque le chiffre "multiplicatif" de la base depasse

*** Definition mathematique :**

Un nombre peut etre decompose en bases. Cela peut paraitre evident car nous acquerons ce reflexe par un apprentissage des l'enfance.

Nous allons decomposer un nombre ,6258 par exemple

6258 = 6 x (1000) + 2 x (100) + 5 x (10) + 8 x (1)
= 6 x (10^3) + 2 x (10^2) + 5 x (10^1) + 8 x (10^0)

C - L'OCTET

L'octet est un nombre de huit bits. Le microproceseur 8080A travaille sur des mots de 8 bits. D'autres plus perfectiones utilisent des mots de 16 bits, voir de 32 bits.

Un nombre de 8 bits peut varier, dans l'absolu, de 0 a 255.
On calcule cette valeur par (2^n)-1, ici n=8 donc 2^8-1=255.
Nous verrons, par la suite que le microprocesseur peut se resever un bit pour le signe.

D - L'HEXADECIMAL

L'hexadecimal correspond a la base 16. Les symboles utilises sont les 10 chiffres et les 6 premieres lettres de l'alphabet. Un paragraphe est consacre a cette base.

2 - LES CONVERSIONS

A - Conversion BINAIRE-DECIMAL

Vous pouvez donc vous appercevoir que pour obtenir la valeur decimale d'un nombre, quelque soit sa base, il faut multiplier chaque chiffre de rang R par (base)^R et additionner le tout (R commencant a 0) .

Il en est de meme pour la base 2, le binaire. Le chiffre correspondant au bit.

Exemple : 1011

nombre.....	1	0	1	1	
rang R.....	3	2	1	0	
2^R	8	4	2	1	base = 2
resultat : chiffre x 2^R	8x1=8	4x0=0	2x1=2	1x1=1	

somme : 8 + 0 + 2 + 1 =11 (decimal)
donc 1011 (binaire) = 11 (decimal)

B - CONVERSION DECIMAL-BINAIRE

L'operation inverse est aussi aisee. Deux methodes :

1 ere methode
- - - - - - -

il faut chercher le plus grand facteur (1,2,4,8,16,32,64,128...) contenu dans le nombre a convertir

exemple : 69 decimal

plus grand facteur	: 64	reste 5
" " "	: 4	" 1
" " "	: 1	" 0

donc 69 = 1x(2^6) + 0x(2^5) + 0x(2^4) + 0x(2^3) + 1x(2^2) + 0x(2^1) + 1x(2^0)

69 = 1 0 0 0 1 0 1

69 = 1000101 (binaire)

2 eme methode
- - - - - - -

Une division successive du nombre par deux permet, en relevant les restes,de retrouver le nombre en binaire

```
6 9 I 2
  1 I----
    I 3 4  I 2
    I   0  I----
           I 1 7 I 2
           I   1 I----
                 I  8  I 2
                 I  0  I----
                       I   4 I 2
                       I   0 I----
                             I   2 I 2
                             I   0 I----
                                   I   1 I 2
                                   I   1 I----
                                         I    0
                                         II
```

Si on lit les restes de bas en haut, on trouve le resultat suivant : 1000101
comparez avec celui trouve plus haut !

C - CONVERSION BINAIRE-HEXADECIMAL

Il est difficile et meme imposible de travailler directement en bits. On utilise alors la base 16 car elle presente la particularite d'etre directement accessible a partir du binaire et vice versa.

pour convertir un octet en hexadecimal, il faut diviser le mot en deux fois 4 bits et convertir chacun d'eux en hexadecimal.

exemple : 10110101 a convertir

valeur binaire :	1011	0101
valeur decimale :	11	05
" hexa. :	#B	#5
resultat :	#B5	

1D	correspond a	#1	HEXADECIMAL
2D	"	#2	"
3D	"	#3	"
.	.	.	.
.	.	.	.
9D	"	#9	"

10D	"	#A	"
11D	"	#B	"
12D	"	#C	"
13D	"	#D	"
14D	"	#E	"
15D	"	#F	"

3 - LES OPERATIONS BINAIRES

Ces operations sont identiques a celles de la base dix. Si vous ne comprenez pas aisement, reportez vous a ces dernieres, analysez les et vous decouvrirez que tout cela est tres simple.

A - ADDITION

Operation sur 1 bit

```
0 + 0 = 00
1 + 0 = 01
0 + 1 = 01
1 + 1 = 10
```

Exemple :

```
(retenues) ...è1 1111 1 é
                 10110101
               + 10101101
               -----------
               = 101100010
                 ^
               Carry : C'est le neuvieme bit. Il indique si le resultat est
                       a 8 bit ( >255 )
```

C - NEGATION

Le signe "-" n'existe pas en binaire. Le huitieme bit indique si le nombre est negatif. Pour rendre un nombre negatif, il faut complementer l'octet a deux. C'est a dire remplacer les 1 par des 0, les 0 par des 1 et ajouter 1 a ce nombre obtenu.

Exemple : -6

```
6 ................     0 0 0 0 0 1 1 0

Complement a 1  ..     1 1 1 1 1 0 0 1
Complement a 2  ..   +               1
                     -----------------
-6 ................  = 1 1 1 1 1 0 1 0
```

D - SOUSTRACTION

La soustraction n'existe pas en binaire. Il faut recourir a une astuce.
A - B = a + (-B)

On sait rendre un nombre negatif, voir ci-dessus.

Exemple : 17 - 6

```
17  ................   0 0 0 1 0 0 0 1
-6  ................   1 1 1 1 1 0 1 0
                       ---------------
17-6  ..............   1 1 1 0 1 0 1 1          = 11 decimal
```

E - MULTIPLICATION

Cette operation est la meme qu'en decimal

```
1 x 0 = 0
1 X 1 = 1
0 x 1 = 0
0 x 0 = 0
```

Exemple : 9 x 11

```
  9  ..............          1 0 0 1
X 11 ..............        x 1 0 1 1
----                  ---------------
  99                        1 0 0 1
                    +     1 0 0 1
                    +   0 0 0 0
                    +  1 0 0 1
                    -----------------
                    =  1 1 0 0 0 1 1      = 99 decimal
```

4 - LES OPERATEURS LOGIQUES

Les operations logiques s'effectuent bit par bit. Lorsque l'on doit utiliser un tel operateur sur un octet, il faut "operer" bit par bit, le 1er bit avec le 1er etc...), sans tenir compte des autres. Voici les principaux :

A - NON (NOT)

Si P = 1 alors NON-P = 0
   P = 1 NON-P = 1

B - OU (OR)

1 OU 1 = 1 C'est l'addition logique
1 OU 0 = 1 La notation P OU Q : P + Q
0 OU 1 = 1
0 OU 0 = 0

C - ET (AND)

1 ET 1 = 1 C'est la multiplication logique
1 ET 0 = 0 La notation P ET Q : P*Q : PQ
0 ET 1 = 0
0 ET 0 = 0

D - NON-ET (NAND)

La combinaison du AND et du NOT
P NAND Q = NON(PQ)

1 NAND 1 = 0
1 NAND 0 = 1
0 NAND 0 = 1
0 NAND 1 = 1

Vous pouvez remarquer que cette operation correspond a (NON-P) OR (NON-Q)

E - NON-OU (NOR)

La combinaison du OU et du NOT
P NOR Q = NON(P+Q)

1 NOR 1 = 0
1 NOR 0 = 0
0 NOR 1 = 0
0 nor 0 = 1

remarquez qu'elle correspond a (NON-P) AND (NON-Q)

F - EXEMPLE SUR DES OCTETS

```
        10110110                10111101
     OU 10010100              ET 01101110
     -----------              -----------
     =  10110110              =  00101100


        10010110                10010100
   NAND 10010010              NOR 10010110
   -------------              ------------
   =    01101101              =   01101001


    P = 10010110
NON-P = 01101001
```

Ceci met fin a la premiere partie d'une suite d'articles concenant le microproceseur. La seconde traitera du microprocesseur lui-meme et de son architecture sans toutefois aller trop loin dans les details, inutiles ici, et que les mordus trouveront dans des livres tres compliques.

HP.LEGRY,
F-59 DOUAI.

CARTE X-BUS

CHOPPINET Eric
27 rue Louis Pasteur
91310 LEUVILLE SUR ORGE
FRANCE
Tel: (6) 084.60.87

Leuville le 4 mars 1985

Cher ami DAIiste,

Je vous remercie de l'intérêt que vous portez à la carte EXTENSION-BUS.

Le délai de livraison de la carte nue réalisée en trous métallisés est de 6 semaines après réception de commande.

Pour l'installation de la carte, il faudra vous armer:
- d'un multimètre,
- si possible d'un oscilloscope,
- d'un grattoir pour couper 3 pistes sur la carte mère du DAI,
- d'un bon fer à souder (panne très fine !!!),
- et de fil à wrapper.

Pour la mise en service de la carte, il est prudent de le faire par étapes.

*** Souder toutes les petits composants
(résistances,condensateurs,diodes,transistors.)
(connecteur X-BUS, supports circuits intégres.)

*** Apporter les modifications suivantes sur la carte mère:

1. Couper la piste arrivant à la pin 47 du X-BUS
2. Couper les pistes arrivant aux pins 22 et 21 du X-BUS et relier ces pistes sans passer par le X-BUS.
3. Relier IC44 pin 9 à la pin 47 du X-BUS (F0XX).
4. Relier IC45 pin 10 à la pin 18 du X-BUS (F9XX).
5. Relier IC45 pin 11 à la pin 22 du X-bus (FAXX).
6. Relier IC107 pin 1 à la pin 21 du X-BUS (reset).

*** Vérifier les modifications à l'ohmmètre.

*** Mettre votre DAI sous tension sans MEMOCOM.

*** Contrôler le + 5V sur chaque support de CI.

*** Introduire dans le bon sens les CI suivants:
- CI 5 74C00 (CMOS ou HCMOS)
- CI 6 74C42 (CMOS ou HCMOS)
- CI 3 votre EPROM DCR.

<<<< COUPER L'ALIMENTATION >>>>

*** Passer en UT. Visualiser les adresses F000-F7FF. Le contenu de la ROM apparait, si ne n'est pas le cas,vous avez mal adressé le décodeur 74C42. (SWITCH A - B - A11 ouverts)

Avec un oscilloscope,vous pouvez visualiser (F000-F7FF)
CI 44 pin 9 (carte mère)
pin 47 du X-BUS
CI 6 pin 12
CI 6 pin 4

*** Brancher votre MEMOCOM, et faire CALLM #F2F2.

Si vous utilisez votre ROM DCR, la compatibilité SOFT est totale.
A ce stade vous ne pouvez plus commuter vos EPROM à l'aide des minirupteurs
Faites POKE #296,0. Maintenant modifiez la position des SWITCH, visualisez les adresses F000-F7FF.
Vous pouvez donc changer de banc mémoire mais vous n'avez plus les commandes DCR.

Vous pouvez introduire le CI 4. (INS 8154 N)

Les SWITCH ne servent qu'à l'initialisation du bon banc mémoire lors d'un départ à froid, ensuite le 8154 prend les commandes en placant les pins du port B en sortie basse impédance. (Par convention, A11 de CI 3 = 5V.)
La ROM DCR a été entièrement réécrite avec des commandes supplémentaires:

RBP Restore Basic Pointer
SBP Save Basic Pointer
IMP initialisation imprimante parallele.
(...)

A l'initialisation le 8154 prend les commandes. A,B,A11 en basse impédance

La nouvelle ROM DCR permet toutes les fonctions:
<LOOK> <LAST> <VER> <CHECK> <LOAD> <SAVE> <CASx> <DCRx>
Il est possible que dans certains programmes on teste la présence de l'eprom DCR par controle d'un byte dans le banc mémoire F000-F7FF. Ce byte risque à présent d'être different.

Concernant le programmateur d'eprom, je tiens à votre disposition le soft correspondant pour des eproms 2732 en langage machine que vous lancerez par un CALLM #7000

** Répondez au menu **
L Lecture en #4000 - #4FFF
V Vérification
T Test de virginité
P Programmation avec données en #4000
F Fin : retour au moniteur

Attention à la tension de programmation (+ 21 V ou + 25 V). Cette tension de programmation doit etre mise au bon moment, (le soft vous previendra et la suprime dès que le programme vous rend la main. Utiliser une alimentation +30 V; le régulateur LM137 (T1) et le potentiometre (P1) permettent d'obtenir 21 ou 25V. (Un essai sans EPROM est conseillé.)

Pour ce qui concerne l'horloge temps réel,le soft est en développement. Le CI 9 MM58174A est adressé de FAC0 à FAFF. Reportez-vous à la notice du constructeur. Le QUARTZ est de 32,768 kHz (standard horlogerie). L'horloge continue à fonctioner même en cas de coupure secteur, je compte sur vous pour me présenter des applications originales.

Un mot sur l'utilisation de la RAM de 2K.(6116)
Pour écrire dans l'espace mémoire F000-F7FF, il faut faire un POKE #10,#C9 pour éviter un STACK OVERFLOW et prendre garde que le SWITCH 'WR' soit fermé, ensuite faite POKE #10,0.
En ouvrant le SWITCH 'WR', vous protégez le contenu de la RAM en cas de coupure secteur.Cette RAM sert à lancer des programmes que vous voulez vérifier avant de les écrire en ROM et non de stockage des DATA. Pour stocker des variables, utilisez la RAM du INS 8154N.

Tous les SOFT se rapportant a cette carte ont ete développés par mon ami Claude PICARD.

Vous trouverez dans les prochains numero de DAI d'autres renseignements sur cette carte.

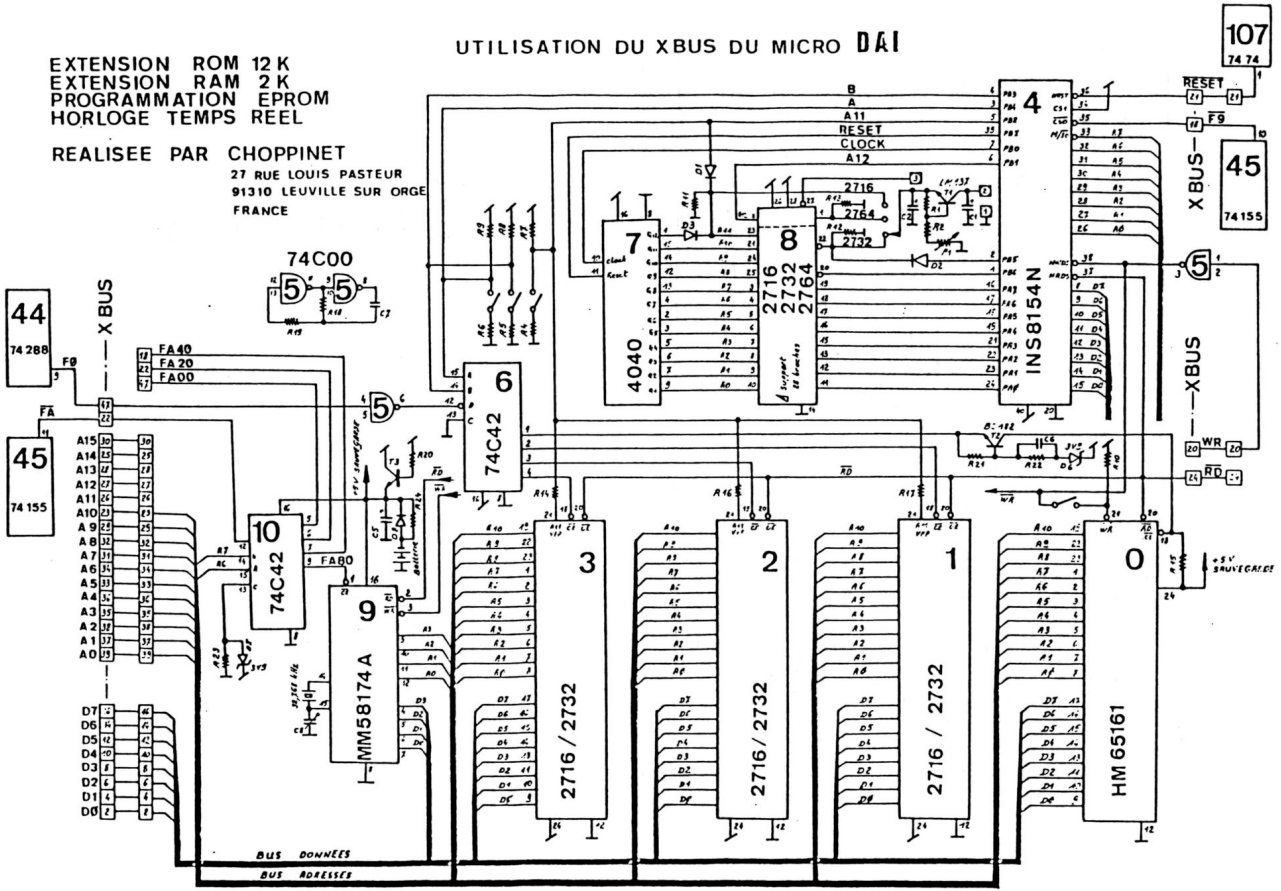
Amicalement.

Eric CHOPPINET

UTILISATION DU XBUS DU MICRO DAI

EXTENSION ROM 12 K
EXTENSION RAM 2 K
PROGRAMMATION EPROM
HORLOGE TEMPS REEL

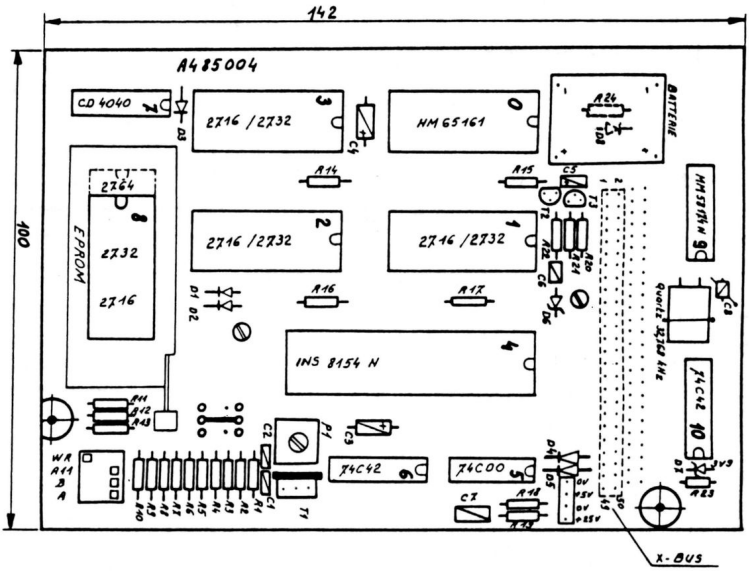
REALISEE PAR CHOPPINET
27 RUE LOUIS PASTEUR
91310 LEUVILLE SUR ORGE
FRANCE



Note 1 — Modification carte mère

1. Couper la piste arrivant à la pin 47 du x-bus
2. Couper la piste arrivant à la pin 22 et 21 du x-bus et relier les pistes sans passer par le x-bus.
3. Relier ICC 44 (9) à pin 47 du x-bus (F000 - F7FF)
4. Relier IC 45 (10) à pin 18 du x-bus (F300 - F9FF)
5. Relier IC45 (11) à pin 22 du x-bus (FA00 - FAFF)
6. Relier IC 107 (4) à pin 21 du x-bus (Reset)

Note 2 — Avant d'écrire dans la RAM (F000 - F7FF)
Faire : poke # 10, # C9
pour éviter "stack OVERFLOW"



LISTE DES COMPOSANTS

Résistances

R1 ..... 100 Ω
R2 ..... 1,5 kΩ
R4 ..... 470 Ω
R5,R6,R7..... 4,7 kΩ
R8,R9 ..... 47 kΩ
R10 ..... 4,7 kΩ
R11,R12,R13 .. 47 kΩ
R14 ..... 1 kΩ
R15 ..... 4,7 kΩ
R16,R17 ..... 1 kΩ
R20 ..... 4,7 kΩ
R21 ..... 10 kΩ
R22 ..... 470 Ω
R23 ..... 10 kΩ
R24 ..... 100 Ω

Condensateurs

C1,C2...1 µF – 10 V (Ta)
C3,C4...10 nF
C5 .....1000 µF – 16 V
C6 .....100 nF
C8 .....10 pF ou 30 pF ajust.

Semi-conducteurs

D1,D2,D3 .....1N 4148
D6,D7 .....BZX 3,9
D8 .....OA 81 (Ge)

T1 .....TDB 0117
T2,T3.....BC 548
IC0..... 6116
IC1,IC2,IC3 2716
.....2732
.....2764
IC4.....INS 8154 N
IC5.....74 C 00
IC6,IC10 .....74 C 42
IC7.....4040
IC8.....*EPROM vierge*
IC9.....MM 58174 A

Divers

Support à insertion nulle (28 br.)
Inter DIL (4 inters)
Connecteur 50 broches
(2 x 25; pas 2,54)
Quartz 32,768 kHz

PETITES ANNONCES

PETITES ANNONCES

Vends DAI + 1 paddle + CLIO + pgms divers + mode d'emploi français ou ECHANGE contre imprimante (SEIKOSHA, EPSON) valeur équivalente. D'ARANTES Luc, Saint Pancré, F-54730 GORCY.

Vends DCR achat début 83 + une 40aine de K7: 2000 FF état neuf ou 2500 FF avec K7. TRAN QUY, rue René LEFEVRE 52, F-41000 BLOIS, T.(16-54) 43-68-27.

Cherche MEMOCOM + TOS d'occasion. Cherche DAInamic janvier '84. Philippe VERHAEGE, rue BEL AIR 5, B-7190 ECAUSSINE.

Vends cassettes CERTIFIEES digitales pour DCR: 1600 FB les 6 (port compris); cassettes DCR non-certifiées: 700 FB les 6 (port compris); EPROM NEW CHARACTERS GENERATOR: 500 FB. Christian POELS (IDC).

Carte PAL 3000 FB (à discuter); TOS DCR 900 FB. Marc VANDERMEERSCH (IDC).

REMARQUES: Ces petites annonces gratuites pour les abonnés sont exclusivement réservées à des propositions entre particuliers sans objectif commercial et relatives à l'informatique. DAICLIC se réserve le droit de refuser une annonce sans avoir à fournir de justification.

Envoyez vos annonces à la rédaction:

I.D.C. - DAICLIC
C/O Christian POELS
Rue des Bas-Sarts 10
B-4100 SERAING

News

DE NOTRE ENVOYE SPECIAL AUX ETATS-UNIS:

La société hollandaise SILO PROJECT a installé un système de controle technique de production géré par un DAI à l'usine de General Electric d'Albani aux Etats-Unis.

Le DAI réalise une analyse constante de contraintes mécaniques de la composition chimique...

U N E S O U R I S
P O U R V O T R E D A I .

par Dominique CARLIER (DAIC).

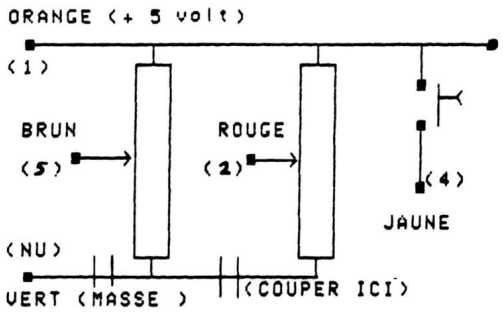
En tant qu'utilisateur du DAI j'ai toujours cherché les moyens me permettant d' utiliser le DAI comme machine utile. Malheureusement, les manettes du DAI ne convenant pas pour toutes les applications, j'ai opté pour une solution qui me paraissait comme étant meilleure. C' est ainsi que je me suis procuré une 'color mouse' de chez Tandy. Après un rapide coup d'oeil il s'avère qu'il faut faire quelques adaptations à l'engin en question pour pouvoir l' utiliser convenablement avec le DAI.

Tout d'abord il faut changer le connecteur. Ensuite, il y a une piste à couper dans la souris. La figure 1 donne le schéma électrique de la souris et les couleurs des fils du connecteur (telles que je les ai trouvées dans mon exemplaire). Entre parenthèses vous trouverez les numéros correspondants du connecteur DAI.

Une fois ces modifications opérées, la souris se connecte sur une entrée paddle et s ' emploie comme celle-ci.

****
* figure 1 *
****

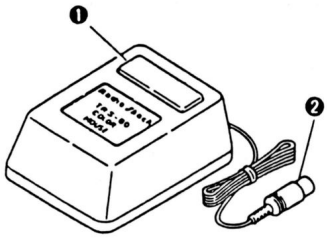
TRS-80 COLOR MOUSE SUR DAI-PC



La figure 2 vous donne une idée de l'aspect de cette souris et à la figure 3 vous trouverez les différents mouvements possibles. Voici également quelques caractéristiques techniques :

déplacement maximal		127 mm
longueur du cable		1,5 m
dimensions :	longueur	90 mm
	largeur	62 mm
	hauteur	34 mm
poids		210 g

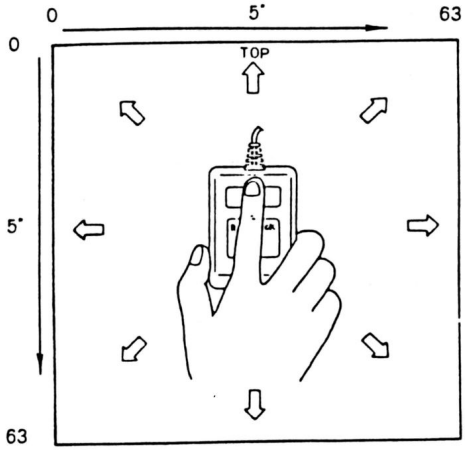
****
* figure 2 *
****



Cette souris (réf 26-3025) valait , en 1984, 2295 francs et n'est donc pas des moins chères. Si vous ne désirez que des manettes à 2 dimensions avec bouton poussoir il faut signaler les autres produits Tandy :

- color computer joystick ref 26-3008 à 1295 f la paire
- precision joystyck ref 26-3012 à 1295 f la pièce.

****
* figure 3 *
****



Si vous comptez adapter une souris comme décrit précédemment ces notes vous seront très utiles.

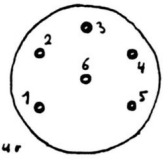
Quelques renseignements :

- La souris se branche dans la sortie PDL 1 ou PDL 2. Elle vous donnera tout comme un potentiomètre ou un joystick une valeur précise allant de 0 à 255 en X & Y.
- La souris est totalement compatible avec les logiciels utilisant les paddles , et ce sans aucune modification de ces logiciels. Certes l'utilisation de la souris avec certains programmes rend ceux-ci difficile d'emploi mais il en est d'autres où elle est particulièrement bien adaptée, tel le programme EDITEUR DE DESSINS.
- Nous vous conseillons d'utiliser la souris sur une surface plane , de préférence sur une feuille de papier pour avoir une certaine adhérence.
- Il est conseillé de ne pas toucher la bille avec les doigts et de nettoyer (dégraisser...) cette bille régulièrement.

Adaptation :

Enlever la fiche livrée avec la souris (5 broches) et souder les fils , une fois dénudés , dans une nouvelle fiche (6 broches males) comme sur la figure a.

```
************
* figure a *
************
```



vue de l'intérieur

broches	-	couleurs
1	-	ORANGE
2	-	ROUGE
3	-	
4	-	JAUNE
5	-	BRUN
6	-	VERT

Retourner la souris (bille vers le haut) et dévisser les 3 vis (aux extrémités) qui tiennent le 'capot'.
Ouvrir délicatement en veillant à garder la bille vers le haut. Poser le capot avec la bille et le bouton EVENT (rouge) en prenant garde de ne pas égarer le petit ressort.

Dévisser le circuit imprimé afin de pouvoir couper la piste venant de la masse (fil VERT) aux deux endroits décrits sur la figure 1 (voir plus haut).
Une fois cette opération effectuée, remonter le circuit imprimé.

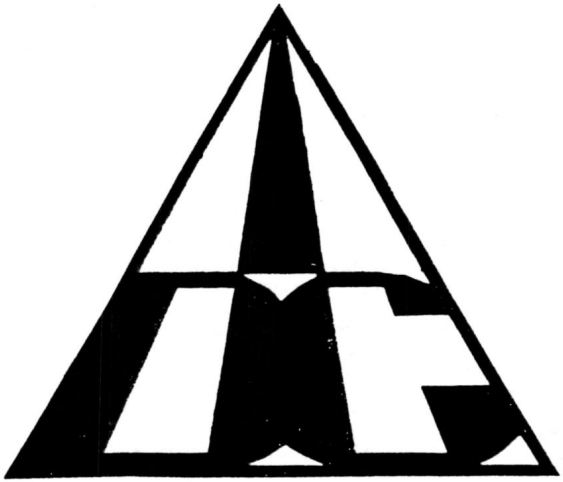
Inverser les fils noir et rouge situés aux extrémités du potentiomètre mesurant les mouvements verticaux.

Remonter le tout en prenant garde d'une part de ne pas trop toucher la bille avec les doigts, d'autre part de ne pas briser la petite tige supportant le ressort.

Et maintenant à vos programmes.

Fabrice DULUINS pour D.A.I.C.

```
1       REM ************************************
2       REM *  DEMO  SOURIS  EN  PAGE  TEXTE  *
4       REM * (c) Alain Mariatte & DAIclic 85 *
5       REM ************************************
6       GOSUB 1000
7       AD%=#BFE7:X%=0:Y%=0:P%=32:GOTO 55
10      X%=PDL(0):Y%=PDL(2)

11      REM SELON VOTRE VERSION DU DAI CHANGER Y%=PDL(1)
20      X%=X%/4:Y%=Y%/11:IF X%>59 THEN X%=59
21      IF PEEK(#FD00) IAND 32=32 THEN 200
22      IF XX%=X% AND YY%=Y% THEN 10
23      POKE AD%,P%:POKE AD%-3,0
25      CURSOR X%,Y%:AD%=PEEK(#73)*256+PEEK(#72):P%=PEEK(#77)
30      PRINT CHR$(P%);:POKE AD%-3,#FF
33      CURSOR 4,23
55      XX%=X%:YY%=Y%
100     WAIT TIME 1:GOTO 10
200     SOUND 0 0 15 0 FREQ(440.0):WAIT TIME 10:SOUND OFF
210     G%=GETC:G%=GETC:G%=GETC
211     G%=GETC:IF G%=0 THEN 211
220     POKE AD%,G%:GOTO 25
1000    MODE 0:COLORT 8 0 10 0:PRINT CHR$(12)
1001    POKE #8A,#69:POKE #8B,#BF
1010    POKE #8A,#69:POKE #8D,#BF
1020    POKE #8C,#E5:POKE #8D,#B3
1030    POKE #84,#D5:POKE #85,#B3
1040    POKE #8E,#D5:POKE #8F,#B3
1050    PRINT CHR$(12):LIST 1-220
1060    RETURN
```

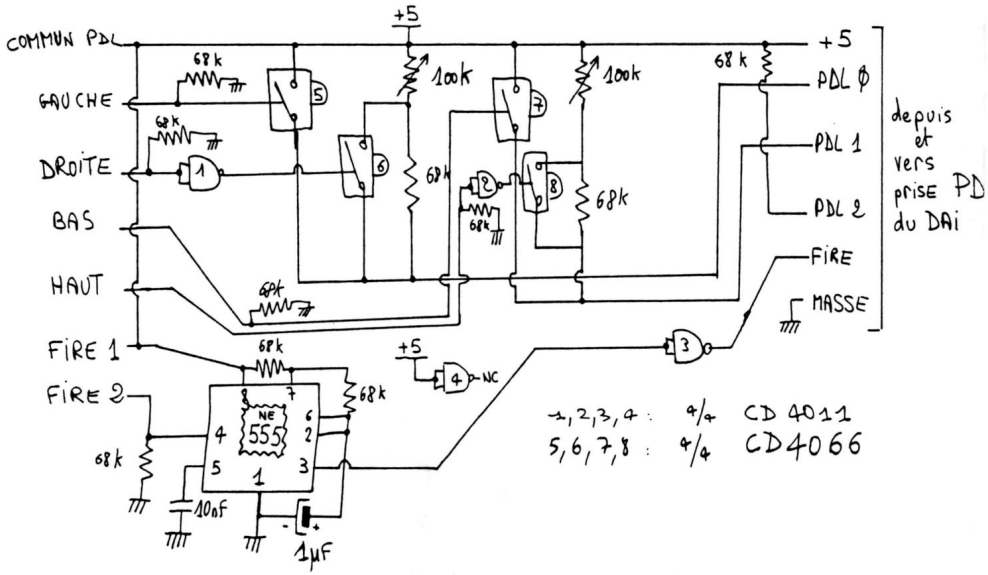


INTERFACE POUR JOYSTICKS ATARI, SPECTRAVIDEO, ...

Vous qui vous êtes toujours plaint de ne pouvoir profiter de véritables mannettes de jeu pour votre DAI, celles-ci étant vendues à un prix exhorbitant par Multisoft (sans pub.), ce montage vous concerne.

En effet, ayant pendant un temps rongé mon frein comme vous, j'ai franchi le pas et créé une interface de faible coût mais qui vous ouvre l'énorme marché des joysticks fonctionnant sur ATARI, CBM 64, consoles de jeux,... tous ces joysticks possèdant (c'est rare) une fiche de sortie (type CANON 9 broches) cablée universellement de la même façon.

Après ce petit préambule, je vous décrit la 'bestiole' en question: une petite boite, grosse comme une boite d'allumettes grand-format, qui se charge de tout l'interfaçage nécessaire pour que notre cher DAI puisse ingurgiter ce que le joystick va lui fournir. De plus, pour les jeux d'action, une fonction 'répétition' pour le tir est incluse afin d'éviter de maltraiter le bouton de tir du joystick lors d'une attaque particulièrement farouche dans un épisode de DAYLAXIANS, PACMAN ou autre.



Les fanatiques d'électronique reconnaitront aisément 3 circuits intégrés très classiques : un CD 4011 (4 portes NAND-CMDS), un CD 4066 (4 interrupteurs CMDS) et un NE 555 (multivibrateur).

Le NE 555 sert pour la répétition (AUTOFIRE pour les anglophones) du tir. En effet, la broche 4 (RESET) de ce circuit est reliée au bouton de tir du joystick. Lorsque le tir est relaché, un niveau 0 est présent sur cette broche et maintient à 0 la sortie du NE 555 en broche 3: pas de tir. Lorque le tir est activé, la broche 4 passe à 1 (+5V) et le NE 555 se met à osciller en délivrant une rectangulaire de 7 Hz environ en broche 3: le DAI va détecter successivement : 'bouton appuyé','bouton relaché','appuyé','relaché',...
Plus besoin d'appuyer frénétiquement sur le bouton du joystick: un simple appui permanent suffit. Naturellement, le bouton relaché, le NE 555 s'arrete jusqu'à la prochaine sollicitation.

La détection des mouvements se fait comme suit :

- paddle à gauche	PDL(0) = 0
- paddle à droite	PDL(0) = 255
- paddle au milieu	PDL(0) = 128
- paddle en haut	PDL(1) = 255
- paddle en bas	PDL(1) = 0
- paddle au milieu	PDL(1) = 128

Seules ces 6 positions de la mannette sont accessibles (avec les 4 combinaisons 2 à 2).
Le joystick sollicite un contact lorsqu'il atteint chacune de 4 positions extremes (haut-bas-gauche-droite); tous les contacts sont relachés lorsqu'il est en position médiane.

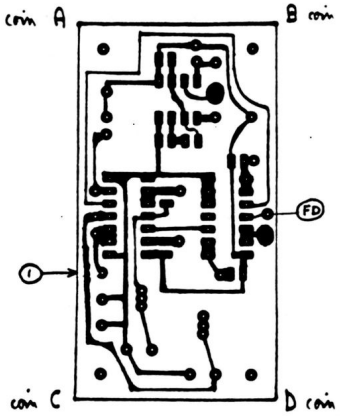
Le CD 4066 et le CD 4011 se chargent de détecter puis d'intepréter ces informations en connectant ou en courcircuitant les résistances comme suit:

- gauche	: liaison directe avec +5V	
- droite	: RV1 + R1	sur PDL(0)
- milieu horizontal	: RV1	
- bas	: liaison directe avec +5V	
- haut	: RV2 + R1	sur PDL(1)
- milieu vertical	: RV2	

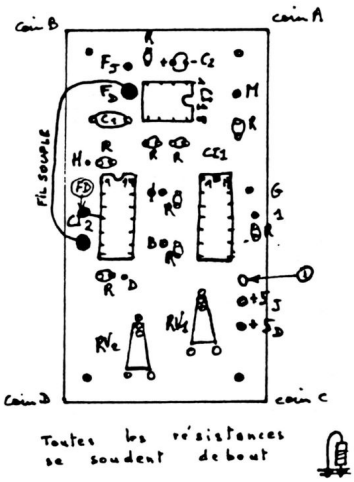
En sortie , on obtient des niveaux variables sur PDL(0) et PDL(1) en fonction de la position du joystick. Le décodage est immédiat et infaillible !

CIRCUIT IMPRIME :

Circuit imprimé coté cuivre

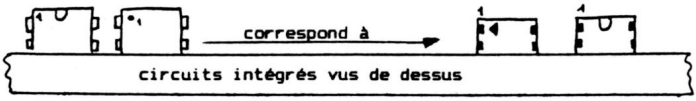


Circuit imprimé coté composants



Peu de remarques à faire. Le circuit imprimé peut-être reproduit soit par transferts + gravure au perchlorure de fer, soit par photo + gravure au perchlorure (plus couteux), soit par une entreprise spécialisée qui se chargera du travail moyennant finance et le plan du circuit (quand même) face 'cuivre'.

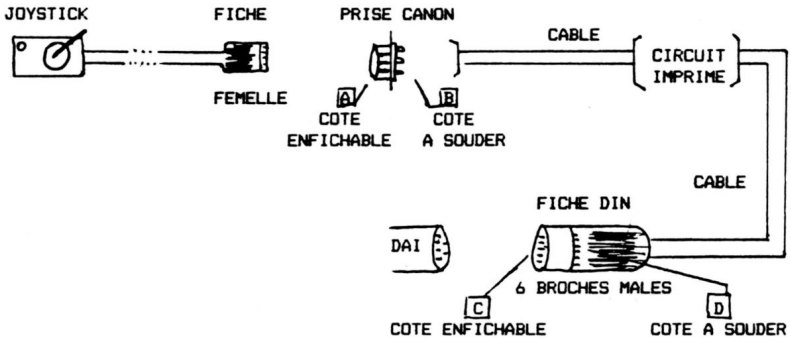
Une fois le circuit percé (et étamé éventuellement), on procèdera à la soudure des composants : d'abord les résistances (on ne peut se tromper : elles ont toutes la MEME valeur) puis les 2 condensateurs puis les supports des 3 circuits intégrés (je vous le conseille fortement). On enfichera pour finir les 3 circuits intégrés dans leurs supports respectifs et DANS LE BON SENS.



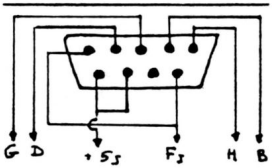
On peut maintenant passer au ...

CABLAGE :

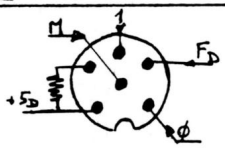
La plus grosse partie du boulot.
Pour commencer, je définis le nom des différentes fiches que je vais employer:



COTE B DE LA PRISE CANON



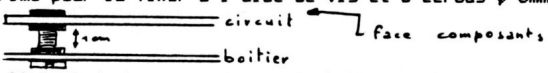
COTE D DE LA FICHE DIN 6 BROCHES



Soignez les soudures (résistance accrue à la traction). Pour les connections, utilisez du fil 1 conducteur de faible.
Pour le cable qui relie circuit imprimé et fiche DIN, il n'est pas du tout nécessaire d'utiliser du cable blindé 4 conducteurs + masse (rigide et couteux). On pourra utiliser du cable en nappe 5 conducteurs qui a l'avantage d'etre très souple et d'un prix presque ridicule.

MISE EN BOITIER :

Tout boitier plastique peut convenir. Avec tout le soin nécessaire, on percera :
- 1 trou pour le cable vers la DIN 6br.
- 1 trou trapézoidal pour la prise CANON
- 4 trous sur le fond du boitier correspondant aux 4 coins du circuit imprimé pour la fixer à l'aide de vis et d'écrous ∅ 3mm



Le boitier va faire l'objet de beaucoup de manipulations plus ou moins brutales; il est donc important de bien fixer le circuit imprimé.

ESSAI ET REGLAGES :

Deux réglages minimes sont nécessaires.
Branchez la fiche DIN dans la prise PDL 1 du DAI et branchez votre joystick dans la fiche CANON.
Allumez votre DAI, vérifiez qu'aucun des circuits intégrés ne chauffe : s'il y en a un qui semble faire office de 'radiateur', débranchez tout et revérifiez votre cablage (le circuit intégré 'radiateur' peut avoir claqué...)

Si ce cap est passé, tapez en BASIC :

```
10   PRINT PDL(0),PDL(1): GOTO 10  <return>
```

faites RUN <return> et ne touchez pas au joystick (mannette en position médiane).
Ajustez RV1 de façon à ce que l'on ait 128 (ou éventuellement 127) comme première valeur affichée à l'écran.
Faites de meme avec RV2 pour avoir 128 comme seconde valeur.
Ceci terminé, tapez NEW <return> puis UT <return>

```
>Z3
>S300  xx-3A  xx-00  xx-FD  xx-E6  xx-20  xx-CD  xx-1D  xx-ED
       xx-3E  xx-0D  xx-EF  xx-03  xx-C3  xx-00  xx-03
>G300
```

Le DAI affiche toutes les valeurs récoltées au niveau du bouton de tir.
Vérifiez qu'à l'écran, lorsque le bouton est relaché, n'apparaissent que des 00 et lorsque le bouton est appuyé, apparaisse une succession de 20 et de 00.
Si tel n'était pas le cas, revérifiez votre cablage.

Repassez en BASIC et retapez la ligne BASIC vue plus haut et manoeuvrez le joystick tout en vérifiant les valeurs affichées qui doivent être :

	1ère valeur	2ème valeur
à gauche	0	128
à droite	255	128
en bas	128	0
en haut	128	255
en bas-à gauche	0	0
en haut-à gauche	0	255
en bas-à droite	255	0
en haut-à droite	255	255

Et voilà ! refermez le boitier de l'interface ... et réjouissez-vous ! Partez à l'assaut de vos jeux préférés que vous allez découvrir d'un jour nouveau: le PAC-MAN devient plus facilement maniable à présent et je suis sûr que votre ancien record ne tiendra pas le coup très longtemps...

LISTE DES COMPOSANTS :
---

1 x CI1 : CD 4066 (ou CD 4016 : meme brochage, ou HCF 4066 ou HEF 4066)
1 x CI2 : CD 4011 (ou HCF 4011 ou HEF 4011)
1 x CI3 : NE 555 (ou équivalent, demandez conseil à votre revendeur)
10 x R : résistance 68 kΩ - 1/4 W (5 ou 10 %)
1 x C1 : condensateur 10 nF - 63 V non polarisé
1 x C2 : condensateur ou tantale 1µF - 10 V minimum
NB vérifiez que la taille des condensateurs que vous achetez permettent leur implantation sur le circuit imprimé
2 résistances ajustables couchées 100 kΩ
1 joystick type ATARI, SPECTRAVIDEO (1 ou 2),... (muni d'une fiche CANON femelle)
1 fiche CANON male chassis 9 broches
1 boitier plastique aux dimensions adéquates
2 supports de circuit intégré 14 broches } à souder
1 support ' ' ' 8 broches }
du cable 1 fil et du cable nappe 5 fils

ATTENTION : CI1 et CI2 sont ultra-sensibles aux décharge d'électricité statique. Ne les touchez pas avec les doigts et conservez les enrobés de papier aluminium jusqu'à ce que vous les insériez dans leurs supports.

NOTE : si vous ne voulez plus avoir à retoucher périodiquement RV1 et RV2 pour rectifier les réglages, immobilisez les curseurs de ces 2 résistances variables à l'aide d'une simple goutte de cire de bougie fondue. Bien que peu recommandée en Haute Fréquence, cette 'astuce' vous sera d'une grande utilité face aux chocs que va subir votre interface...

(c) Pascal JANIN le 31 décembre 1984

ASSEMBLEUR

****
* PROTECTION ET DEPROTECTION : une enquete de DAICLIC *
****

Comment déplomber TOTO
diffusé par le CCC
(Computer Casseurs Club)
(cfr: DAIlassement !)

par

HP LEGRY

F-59 DOUAI

Vous avez sans doute rencontré des programmes protégés que vous auriez voulu dupliquer, ne serait-ce pour les mettre à l'abri de fausses manoeuvres. Avec un peu d'imagination et le FIRMWARE MANUAL, vous pourrez copier ces logiciels pour, je vous le conseille fortement, votre usage personnel.

Ne vous attendez pas à trouver des 'recettes' pour chacun des programmes protégés. Ce n'est pas le but de ce qui suit.
Je vais essayer de vous expliquer une méthode de protection employée dans quelques programmes...

Nous allons prendre un exemple (puisqu'il faut un exemple) d'un programme protégé, diffusé par le CCC (Computer Casseurs Club), le programme TOTO (restons originaux !)

TOTO se compose d'un loader et d'un programme protégé. Il nous faut donc charger cette partie protégée en évitant le loader, tout en neutralisant l'autostart.

1 - le LOADER

Le loader (ou le USER pour certains programmes) constitue le point faible de la protection. Il indique nécessairement la méthode de chargement de la partie principale. Il est généralement illisible. Celui que nous allons demonter fait partie du 'vieux systeme'. La nouvelle methode, que j'ai rencontrée récemment, est plus efficace. Je ne l'aborde pas, un seul sujet à la fois !

Pour accéder au loader, petit programme LM chargé en mémoire RAM, il faut le lire, le faire tourner 'à vide' puis le stopper. Donc :

Positionner la cassette sur le Loader.
Charger le Loader (LOAD).
Ouvrir et retirer la cassette.
Faire tourner le loader (RUN).
Stopper le loader (BREAK).

Sur DCR vous appuyez sur BREAK tout en refermant le DCR avec une microcassette à l'intérieur.

Maintenant, vous pouvez observer le loader en examinant la pile (#F800-#F900). C'est à cet endroit, en général, que les loaders sont stockés.

2 - Lecture de la partie protégée

Vous avez sans doute lu mon article sur le READ. Les logiciels se présentent sous leur forme langage machine car le load ne permet pas, simplement, de charger un programme entre #0 et #29B (autostart).

a- ouverture du fichier ----------

Un nom classique se compose du type du programme (#31) puis du nom en caractères ASCII. Subtilité, pour certains programmes, le type du programme est remplacé par un caractère non reconnu par un read ou un load, c'est-à-dire différent de 1 (lm =#31) et de 0 (basic = #30).

Ici, pour TOTO ,le code ascii représentant le type de programme est #20 (un espace) .

Donc pour ouvrir le fichier :

```
        MVI B   20H                 ;TYPE DE PROGRAMME = SPACE
        MVI C   0FFH                ;FF POUR L'AFFICHAGE
        LXI H   NAME                ;HL POINTE SUR LE NOM
        .
        .
        .
NAME    DB      4H                  ;NOMBRE DE LETTRES
        BD      'TOTO'              ;NOM DU PROGRAMME ATTENDU
```

Pour plus de sécurité, on pointera H sur 0H. On ne testera pas le nom du programme.

b - lecture du(des) bloc(s) ----------

Un programme machine (WRITE-READ) comporte deux blocs.
Un premier bloc, de deux octets,indique l'adresse ,dans la RAM où le second bloc doit etre charge en mémoire lors de la lecture.

Prenons un exemple , le FGT (#300-#900) :

le premier bloc de deux octets : 0300H
le deuxieme bloc : le FGT

Si le programme n'a pas cette structure, il ne peut etre lu correctemment par un READ .

TOTO se divise en deux blocs. Un bloc de #25A5 a #75F0 puis un bloc de 0 a #25A4 . Chaque partie doit etre chargée aux adresses que je vous ai donnees. Le DAI ne connait pas ces adresses. Contrairement au READ où l'adresse de chargement est fournie par les deux octets formant le premier bloc, il faut indiquer ces adresses par le loader. Par défaut, ces parties seront chargées à partir de 0H, ce qui n'aura pas de sens pour le programme.
Vous allez me demander comment j'ai obtenu la structure du programme sur la cassette : en examinant le loader.
Sachant que l'ouverture du fichier s'effectue par un CD 2CEH et que la lecture de bloc par un CD 2D1H, qu'il faut indiquer ces adresses de chargement dans HL (voir articles précédents).

Structure du programme sur cassette

type : (space) #20
nom : (TOTO) T.O.T.O (en ascii) 1er bloc : #25A5 à #75F0 2eme bloc : #0 à #25A4

Nous allons donc simuler le loader.

```
        ORG     0A000H
```

```
LXI B   20FFH      ; TYPE DE PROGRAMME =' '
LXI H   0H         ; PAS DE NOM ATTENDU
CALL    2CEH       ; OUVERTURE DU FICHIER ;
LXI H   25A5H      ; ADRESSE DE DEBUT 1er BLOCK
LXI D   0A00H      ; FIN DE LA RAM DISPONIBLE
CALL    2D1H       ; LECTURE DU PREMIER BLOCK ;
LXI H   0H         ; ADRESSE DE DEBUT 2nd BLOCK
CALL    2D1H       ; LECTURE DU SECOND BLOCK ;
CALL    2D4H       ; FERMETURE DU FICHIER
RET
END
```

Le second bloc est autostart. Il place l'adresse de lancement en #2D5. Vous avez donc chargé TOTO mais il démarre et vous n'avez pu récupérer la main.

3 - L'AUTOSTART

Pour neutraliser l'autostart, il suffit de décaler les blocs de (au moins #300) #1000 par exemple, ceci afin de ne pas modifier l'adresse #2D5 lors du chargement du programme (voir mon article sur l'autostart).

```
ORG     0A000H
LXI B   20FFH
LXI H   0H
CALL    2CEH
LXI H   35A5H          ; 25A5H+1000H=35A5H
LXI D   0A000H
CALL    2D1H
LXI H   1000H
CALL    2D1H
CALL    2D4H
RET
END
```

Votre programme (0 a #75F0) est chargé, décalé de #1000 (entre #1000 et #85F0). Pour le faire tourner il faut donc le redécaler et le lancer.

a - le décalage

Deux solutions : Le sauvegarder de #1000 à #85F0 sur cassette et le relire avec un offset de (-#1000) soit #C000. Donc le charger par :

```
*UT
>RC000
```

Le programme se charge et démarre tout seul. Il se comporte comme l'original à la seule différence que vous disposez d'une version non protégée.

La seconde solution consiste a programmer le décalage et le lancer par un Call en lm.

Programmation du décalage :

```
DI                 ;DECONNECTE LES INTERRUPTIONS
LXI H   1000H
LXI D   85F0H
LXI B   0H
```

```
CALL    0DE4FH     ; #1000-#85F0 --> #0-#75F0
EI                 ; AUTORISE LES INTERRUPTIONS
```

b - le lancement

L'adresse contenue en #2D5-#2D6 soit #12D5-#12D6 dans la Ram après chargement vous indique l'adresse de lancement.
Il vous faut commuter la banque 0 car le programme ne le fait pas toujours.
Si vous observez ces adresses vous pouvez voir en #12D4, à la place de CD 1F F3 : CD E1 65. C'est l'adresse de lancement.

Donc pour lancer TOTO (de #1000 à #85F0), il faut utiliser le lanceur que vous pouvez inclure à TOTO en le plaçant juste après le programme (en #85F0 par exemple) :

```
DI
LXI H   1000H
LXI D   85F0H
CALL    0DE4FH
EI
MVI A   0H
CALL    0D806H          ;COMMUTE LA BANQUE 0
CALL    65E1H
```

Conclusion
****

Vous disposez d'une copie de TOTO que vous pouvez dupliquer. Gardez-la pour vous et n'en faites pas le commerce...

En résumé, si vous arrivez à examiner le loader, qui est la clé de toute protection (de celles que j'ai rencontrées), vous observez le CD 2CEH et vous obtiendrez le nom et le type de programme. En regardant les CD 2D1H vous decouvrirez le nombre de blocs et leurs adresses. Le plus delicat : l'adresse de lancement est généralement contenue en #2D4.
Attention, les interruptions peuvent etre modifiees. Une d'entre-elles l'est dans TOTO.

Une enquete de LEGRY H-P pour IDC
(une enquete ; pas un piratage !)

N.B : Cet article nous a été envoyé par H.P Legry (vous vous en doutiez), je me suis cependant permis de le modifier à plusieurs endroits. En effet, dans sa version originale, cet article vous expliquait comment déprotéger un programme paru dans le commerce dont je me suis permis de transformer le nom...(pas de chance les pirates !).
La protection décrite ici est un exemple de protection, cela vous donne donc une idée de la démarche du pirate face à elle...
Toutes les protections ayant leurs affinités, il va de soit que ce petit truc ne marche pas avec beaucoup de programmes protégés, ce n'est qu'un exemple...

Marc Vandermeerch

P.S : Quelqu'un a-t-il trouvé de quel programme il s'agissait ? Si oui, qu'il écrive à DAICLIC, une (petite) récompensé l'y attend !

D-BASIC

EXTENDED BASIC (2)

LIBMAT.D constitue une librairie de neuf nouvelles commandes concernant le calcul matriciel. Les noms utilisés dérivent du Basic matriciel standard.

Dans la description ci-dessous A(),B(),C() représentent des matrices préalablement dimensionnées.

-lecture d'une matrice A()	MATREAD A()
-affichage	MATPRINT A()
-mise à zéro	MATZER A()
-mise à un	MATCUN A()
-addition de deux matrices A() B() résultat dans C()	MATADDD A(),B(),C()
-multiplication d'une matrice A() par un nombre M;résultat dans B()	MATSCAL M,A(),B()
-multiplication de deux matrices A() B() résultat dans C()	MATMULT A(),B(),C()
-création d'une matrice identité A() (seule la diagonale principale contient 1,le reste contient zéro)	MATIDN A()
-transposition d'une matrice A() résultat dans B() (inversion des lignes et des colonnes)	MATTRN A(),B()

Voici un exemple d'utilisation après avoir chargé LIBMAT.D

```
1000 DIM A(2,2),B(2,2),C(2,2)
1010 MATREAD A():MATREAD B()
1020 MATPRINT A():MATPRINT B()
1030 MATADD A(),B(),C():PRINT
1040 MATPRINT C()
2000 DATA 1,2,3,4,5,6,7,8
```

La prochaine librairie d'EXTENDED BASIC concernera l'édition de textes avec la commande PRTUSING

Fernand LEMOINE - CLUB CAROLODAI

```
590         REM ****************************************
592         REM EXTENDED BASIC PART 2
594         REM LIBMAT.D LIBRAIRIE :CALCUL MATRICIEL
596         REM Fernand LEMOINE CLUB CAROLODAI
598         REM JANVIER 1984
599         REM ****************************************
600         DEF PROC MATREAD  ARR A
602     1    FOR I=1 TO DIM(A,1)
604     2     FOR J=1 TO DIM(A,2)
606     3      READ A(I,J):
        2     NEXT J:
        1    NEXT I
608         END PROC
610         DEF PROC MATPRINT  ARR A
612     1    FOR I=1 TO DIM(A,1):
        2     PRINT :
        2     FOR J=1 TO DIM(A,2)
614     3      PRINT TAB((J-1)*7);A(I,J);:
        2     NEXT J:
        1    NEXT I
616         END PROC
620         PROCEDURE MATZER  ARR A
622     1    FOR I=1 TO DIM(A,1):
        2     FOR J=1 TO DIM(A,2)
624     3      A(I,J)=0:
        2     NEXT J:
        1    NEXT I
626         END PROC
630         PROCEDURE MATCUN  ARR A
632     1    FOR I=1 TO DIM(A,1):
        2     FOR J=1 TO DIM(A,2)
634     3      A(I,J)=1:
        2     NEXT J:
        1    NEXT I
636         END PROC
640         PROCEDURE MATADD  ARR A,B,C
642     1    FOR I=1 TO DIM(A,1):
        2     FOR J=1 TO DIM(B,1)
644     3      C(I,J)=A(I,J)+B(I,J):
        2     NEXT J:
        1    NEXT I
646         END PROC
650         PROCEDURE MATSCAL M ARR A,B
652     1    FOR I=1 TO DIM(A,1):
        2     FOR J=1 TO DIM(B,1)
654     3      B(I,J)=M*A(I,J):
        2     NEXT J:
        1    NEXT I
656         END PROC
660         PROCEDURE MATMULT  ARR A,B,C
662     1    FOR I=1 TO DIM(A,1):
        2     FOR J=1 TO DIM(B,2)
664     3      C(I,J)=0
666     3      FOR K=1 TO DIM(A,2):
        4       C(I,J)=C(I,J)+A(I,J)*B(K,J):
        3      NEXT K
668     2     NEXT J:
        1    NEXT I:
            END PROC
670         PROCEDURE MATIDN  ARR A
672     1    FOR I=1 TO DIM(A,1):
        2     FOR J=1 TO DIM(A,2)
674     3      A(I,J)=0
676     3      IF I=J THEN
        4       A(I,J)=1:
        3      END IF
678     2     NEXT J:
        1    NEXT I:
            END PROC
```

TEST DAI DOS 1541

On a longtemps dit (et on dit toujours !) que le lecteur de cassettes digitales Memocom est, de par son prix, une excellente alternative entre le système cassette audio et les disquettes sur le DAI.

En effet, cette petite boite noire (ou blanche selon les modèles) 'tourne' très bien ,est d'une fiabilité exceptionnelle (quand on emploie des vraies cassettes digitales) et va tout de meme assez vite. (6000 bauds).

En bref, ses deux gros points noirs sont :

1> le prix élevé des cassettes...

2> le non accès-direct aux fichiers enregistrés...

Depuis peu, nous avons vu apparaitre des drives bizarres autour de notre DAI, des drives de marque Commodore. Vu le succès qu'ils ont actuellement, il fallait bien que je voie 'comment ça marche', ce que je fis lors de mon dernier passage chez notre bien-aimè président C. POELS (je vais monter en grade !) , heureux possesseur de ce système Commodore.

Le système testé est le système hollandais, à ne pas confondre avec le système Commodore allemand.

Tout de suite, j'ai été étonné par le nombre de fonctions du DOS. En effet, on aurait pu s'attendre à un DOS quelque peu faible sur un tel matériel, mais non...

Le système se compose de :

- un interface DAI-Commodore drive.: +/- 6000 FB
- un lecteur Commodore: entre 11300 et 15000 FB suivant l'endroit...

Il faut noter qu'aucune modification n'est à faire sur le drive... (donc si votre fils, frère ou voisin possède un Commodore 64, pas de problème, votre drive est bon pour servir chez eux.)

L'interface est en 2 parties : un petit circuit imprimé style TOS DCR à mettre dans les entrailles du DAI (sur le BUS) et un petit boitier d'interface extérieur.

En fait, le drive Commodore possède d'origine un DOS intégré en ROM. L'interface du DAI ne fait donc que transmettre les commandes à ce DOS, ce qui nous emmène vers le principal problème, bien connu des utilisateurs de Commodore 64 : la vitesse (ou plutot la lenteur !)

Le système ne dépasse pas les 3400 bauds !, la moitié de la vitesse du DCR !

C'est très lent, seulement voilà, il ne faut pas oublier que les drives se placent directement à hauteur du bon programme. Il ne faut pas s'amuser à rebobiner sa cassette et obtenir des choses du style :

REW 1:LOOK (zut, ce n'est pas ce programme là) et de nouveau REW1:LOOK...

N'oubliez pas que comme le DCR rebobine aussi vite qu'il lit, un LOAD"DALIDA" sur disquette Commodore correspond +/- à un REW 1:LOAD"DALIDA" sur DCR... De plus, à la sauvegarde, pas besoin de recherche une place libre sur la disquette puisque le DOS fait ça très bien... Bref, après quelques calculs savant, on perd moins de temps avec le drive COMMODORE qu'avec le MDCR (sans compter la souplesse d'utilisation!!!)

C'est donc à vous de voir si cela vous parait intérressant ou non !

Plus de précisions sur les commandes principales du DOS :

- le DIRECTORY est appellé par DIR. Il affiche à l'écran le nom de tous les programmes sur la disquette (page par page, pas comme les drives INdata qui vous déballent tout en 2 sec et qui, de ce fait, vous empechent de lire le nom de vos programmes stockés sur la disquette...). De plus, un peu comme le Ken Dos, vous pouvez charger un programme directement depuis le directory. En effet, vous avez à votre disposition un petit curseur avec lequel il vous suffit d'aller pointer le programme voulu. Le DAI chargera alors ce programme et le lancera... Inconvénient du directory : quand vous etes passé à la seconde page de directory, vous ne savez plus repasser à la première. Il faut donc refaire un autre DIR.
- Si vous sauvez un programme machine et un programme basic sous le meme nom, il existe alors une commande qui chargera automatiquement les deux programmes, adaptera les pointeurs et lancera la partie BASIC (ex: FGT+BASIC).

- /I <return> : IMP INT
- /F <return> : IMP FPT
- /D <return> : sortie sur écran
- /P <return> : sortie sur imprimante
- /H <return> : affiche l'état de chaque entrée/sortie (on/off)
- /C <return> : rétablit le petit curseur clignotant.
- /M <return> : mode 0
- /... : il y a encore d'autre commandes de ce style, je ne vais pas toutes les citer...
- SCRATCH : efface un fichier
- FORMAT : formatte une disquette.
- COPY : copie un fichier.
- RENAME : renomme un fichier.
- VALID : compacte les fichiers de la disquette, ce qui implique gain de place. (réorganisation du disque, si vous préférez.)
- USER : appel d'une routine machine en mémoire (style PROCEDURE DBASIC)
- AUTOLINENUMBER : numérotation automatique des lignes.

Avec l'interface est fournie une disquette TOOLKIT qui permet, par exemple, de regarder un secteur de la disquette, de charger des programmes Commodore (ce qui n'est pas intérressant, car le Commodore travaille avec beaucoup de POKEs),...

Vous pouvez connecter sans problème 4 DCR et 4 drives en ligne. Il est également possible de connecter une banque d'eproms sur l'interface.

Il est à noter que le système n'accepte que 16 caractères dans les titres des fichiers et qu'il est impossible de faire des fichiers autostart à l'allumage de l'appareil.

Chaque disquette est formattée à 170 K bytes.

En bref :

=========

Un beau petit système qui me semble tout-à-fait capable de détroner notre bon vieux DCR, ce n'est évidemment pas un Ken Dos, mais ce n'est pas mal du tout.

C'est à peine plus cher que le DCR, mais il ne faut pas oublier qu'une disquette coute beaucoup moins cher qu'une cassette digitale.

L'interface Commodore hollandais est complètement compatible DCR.

Marc Vandermeersch
pour le DAIC (Bruxelles)

DAILASSEMENT

Chers amis intellectuels, bien le bonjour !
Dans ce deuxième dailassement, nous allons vous montrez un emploi particulièrement particulier (c'est le cas de le dire !) du DAI:

*** Le DAI, instrument de développement physique ***
(c) IDC / DAIclic / DAIBIL 1985
tous droits réservés

Si vous voulez faire chauffer vos muscles, entrez le petit programme suivant dans votre DAI (enfin par le clavier, vous m'aviez compris!)

```
10 REM BODY BULDAI
20 REM (C) IDC/DAIclic/DAIBIL
30 REM Nous déclinons toute responsabilité quand à tout mauvais
40 REM fonctionnement de votre DAI suite à l'emploi de ce programme.
50 REM
60 REM ### PRESENTATION
70 PRINT CHR$(12):MODE0:COLORT 5 15 5 5:POKE#75,32
80 CURSOR0,17:PRINT" DAI PERSONAL                BOXEUR";
90 POKE#BCCB,#5F:POKE#BC9BM#5F:POKE#BC9A,#40:CALLM#D6DA
100 REM ### PARTIE PRINCIPALE DU PROGRAMME
110 G=GETC:GOTO 110
120 END:REM FIN ### DU PROGRAMME
```

Après avoir tapé ce programme, faites RUN.
Pressez la barre d'espace pour entrer dans le programme principal.

Le but du jeu est de frapper le plus fort sur le clavier, voici la table des scores après emploi de ce programme eprouvant :

- si rien ne se passe, vous etes une mauviette, recommencez...
- si vous obtenez un message d'erreur, vous etes déjà une forte jeune fille, recommencez...
- si vous obtenez un STACK OVERFLOW, vos muscles commencent à chauffer, recommencez...
- si votre DAI se plante (tapisserie, etc....), c'est que vous etes déjà d'une certaine force, mais vous pouvez faire mieux, recommencez...
- si votre ecran se coupe, que votre clavier vole dans l'autre coin de la pièce et qu'une fissure apparait à la surface de votre bureau ou de votre table, vous etes très fort. Dans ce cas, engagez vous au CCC (computer casseurs clubs) .

Des claviers de rechange sont à votre disposition chez tous les bons DEALERS DAI.

Le plus fervent de vos serviteurs, DAIBIL lui-meme !
(mais qui suis-je...???...)

P.S : Cet article a été fait sur un horrible Apple //e (non mais sans blagues hein ???), mais mon DAI est tombé en panne lors de la mise au point de ce programme de musculation...cela vous aidera à savoir qui je suis...

ASSEMBLEUR

```
*************************************************************
*                    READ AND WRITE                         *
*************************************************************
```


D.A.I.C.L.I.C HP & L LEGRY
---------- par
MARS 1985 F-59 DOUAI

Il y a peut, je vous avais propose un article sur la sauvegarde et la lecture de programme sur cassette. Quelques erreurs, dues a mon empressement, s'etaient glissees et compromettaient la reussite des articles qui allaient suivre, surtout ceux concernant la deprotection de programmes (interressant,non !!). Je me devais donc de reecrire cet article, ce que je fais ici. Les programmes ont etes testes.

A - LE WRITE

Lorsque vous sauvegardez un programme en langage machine sur cassette ou DCR (Wladr hadr Name),il est structure de la maniere suivante:

```
                 ==================
                I    Type          I
1er bloc       <I  de programme    I
                I    + nom         I
                 ==================I
                I  Adresse de      I
2eme bloc      <I  debut du        I
                I  programme       I
                 ==================I
                I                  I
                I                  I
                I Programme        I
                I                  I
3eme bloc      <I  en              I
                I                  I
                I  L M             I
                I                  I
                ==================I
```

-Type de programme = caractere ASCII correspondant a 1 (31H) pour le LM
-et a 0 (30H) pour le basic-.
-Nom du programme = chaine de caracteres ASCII = Name
-Adresse de debut de programme = ladr

Exemple: la FGT

Pour sauvegarder la FGT vous effectuez: W300 900 FGT
Vous avez alors: -Type = 31H
-Nom = 'FGT'
-ladr = 300H

Lorsque vous utilisez la commande 'WRITE' le DAI execute les operations suivantes:

-Ouverture du fichier et ecriture du nom du

programme.
-Ecriture de l'adresse de depart.
-Ecriture du programme LM.
-Fermeture du fichier.

Examinons ces operations un peu plus en detail.

a/ Ouverture du fichier

Cette routine se lance par un ' CALL 2C5H '
avec: - HL pointant dans la RAM sur l'adress ou est stocke le nom du programme
- A contenant le type de programme.

Elle demarre le moteur du cassette.
Le nom pointe par HL est structure de la facon suivante:
Le premier octet correspond au nombre de caracteres;les suivants aux aux lettres du nom (code ASCII).
Si on ne desire pas donner de nom,il suffit d'initialiser HL a 0.

b/ Ecriture d'un bloc (adresse de debut et programme LM)

Cette routine se lance par un ' CALL 2CBH '
avec - HL pointant sur le debut du bloc a ecrire
- DE contenant le nombre d'octets du bloc.

c/ Fermeture du fichier

Cette routine se lance par un ' CALL 2CBH '.Elle stoppe le moteur du cassette.

exemple: la sauvegarde d'une FGT

```
        ORG     1000H
;
        MVI A   31H         ;
        LXI H   NAME        ; OUVERTURE DU FICHIER
        CALL    2C5H        ;
;
        LXI H   ADEP        ;
        LXI D   2H          ; ECRITURE DE L'ADRESSE DE DEBUT
        CALL    2CBH        ; (2 OCTETS)
;
        LHLD    ADEP        ;
        LXI D   600H        ; ECRITURE DU BLOC (600H OCTETS)
        CALL    2CBH        ;
;
        CALL    2CBH        ; FERMETURE DU FICHIER
;
        RET
NAME    DB      3H
        DB      'FGT'
ADEP    DW      300H        ; ADRESSE DE DEBUT DE PROGRAMME
        END
```

B - LE READ
****

Comme vous avez pu le voir dans le paragraphe precedent, un programme LM est divise en trois parties.On aura donc 4 commandes a lancer pour un READ:

- Ouverture du fichier et lecture du nom du programme
- lecture de l'adresse de debut du bloc a lire
- lecture du bloc de LM
- fermeture du fichier

Examinons ces commandes plus en detail:

a/ Ouverture du fichier

Cette routine se lance par un ' CALL 2CEH avec:
- HL pointant sur le nom attendu stocke dans la RAM
- C contenant 0FFH pour l'affichage du titre et 0 sinon
- B contenant le type de programme (31H)

Le nom pointe par HL est structure de la facon suivante:
Le premier octet correspond au nombre de caracteres;les suivants aux aux lettres du nom (code ASCII).
Si on ne desire pas tester le nom,il suffit d'initialiser HL a 0.

b/ Lecture de l'adresse de debut programme

Cette routine se lance par un ' CALL 2D1H ' avec:
- HL pointant sur l'adresse memoire RAM a partir de laquelle on desire stocker ces 2 octets
- DE pointant sur l'adresse de la derniere case memoire disponible

c/ lecture du programme LM

Il faut lancer la meme routine (CALL 2D1H) avec:
- HL pointant sur l'adresse memoire a partir de laquelle on desire commencer a stocker le programme
- DE pointant sur l'adresse de la derniere case memoire disponible

d/ Fermeture du fichier

Il faut lancer la routine par un ' CALL 2D4H '

exemple: programme de lecture d'un programme LM

```
        ORG     5000H           ;IL FAUT PRENDRE GARDE A NE PAS SUPEPOSER CE
;                                PROGRAMME AVEC LE PROGRAMME A CHARGER
;
        MVI B   31H             ;
        MVI C   0FFH            ; OUVERTURE DU FICHIER
        LXI H   NAME            ; ET LECTURE DU NOM
        CALL    2CEH            ;
;
        LXI H   ADEP            ;
        LXI D   0F900H          ; LECTURE ET STOCKAGE DANS ADEP DE L'ADRSSE
        CALL    2D1H            ; DE DEBUT
;
        LHLD    ADEP            ;
        LXI D   0F900H          ; LECTURE DU PROGRAMME
        CALL    2D1H            ;
;
        CALL    2D4H            ;FERMETURE DU FICHIER
;
        RET
NAME    DB      3H
```

```
        DB      'FGT'
ADEP    DW      0H
        END
```

C - LE LOAD
****

Le load ,comme je l'ai explique dans le premier article,lit deux blocs :le TEXTBUFFER et la Table des Symboles.
Apres l'ouverture du fichier et la lecture du nom (cf le READ),le LOAD charge le TEXBUFFER a partir de l'adresse contenue en 29FH/2A0H puis la table des Symboles et positionne les pointeurs contenus en 29FH-2A4.

Les routines a lancer sont les suivantes :

a/ Initialisation du HEAP

Cette initialisation est effectuee par un ' CALL 0CB23H '.

b/ ouverture du fichier

Il faut lancer la routine par un ' CALL 2CEH ' avec
- B contenant le type de programme (30H)
- C contenant 0FFH si on desire l'affichage du titre et 0 sinon
- HL pointant sur le nom du programme(cf READ);si(HL)=0 aucun nom n'est attendu et le premier programme basic trouve sera charge

c/ Lecture du TEXTBUFFER

Il faut lancer la routine par un ' CALL 2D1H ' avec
- HL pointant sur l'adresse dans la RAM ou l'on desire comencer a sauvegarder le textbuffer;cette adrésse est lue en 29B/29C
- DE contenant l'adresse de la derniere case memoire disponible

Apres lecture HL contient l'adresse de la fin du bloc+1 et il ne faut oublier de positionner le pointeur correspondant (2A1H/2A2H) par un ' SHLD 2A1H '.

d/ Lecture de la table des symboles

Cette routine est la meme que la precedente;les registres etant positionnes correctement:
- HL contient l'adresse de debut de la table des symboles
- DE contient toujours l'adresse de la derniere position memoire libre

Il faut donc appeler la routine par un ' CALL 2D1H ' et ensuite positionner le pointeur correspondant (2A3H/2A4H) par un ' SHLD 2A3H '.

e/ Fermeture du fichier

Il suffit d'appeler la routine par un ' CALL 2D4H '.

exemple: lecture d'un programme basic

```
        ORG     0A000H          ; ATTENTION:LE PROGRAMME BASIC NE DOIT PAS ECRASER
;                                 CE PROGRAMME ASSEMBLEUR
        CALL    0CB23H          ; INITIALISATION DU HEAP
;
        LXI H   0H              ; PAS DE NOM ATTENDU
        MVI B   30H             ; TYPE DE PROGRAMME
```

```
        MVI C   0FFH            ; AFFICHAGE DU TITRE
        CALL    2CEH            ;
;
        LHLD    29FH            ; ADRESSE DE DEBUT DU TEXTBUFFER
        LXI D   0A000H          ; FIN DE LA RAM DISPONIBLE
        CALL    2D1H            ; LECTURE DU TEXBUFFER
        SHLD    2A1H            ; POINTEUR FIN TEXBUFFER=DEBUT SYMBOL TABLE
;
        CALL    2D1H            ; LECTURE DE LA TABLE DES SYMBOLES
        SHLD    2A3             ; POINTEUR FIN SYMBOL TABLE
;
        CALL    2D4H            ; FERMETURE DU FICHIER
;
        RET
```

D - LE SAVE
****

Le SAVE effectue les operations inverses du LOAD; C'est a dire:

a/ Ouverture du fichier

Cette operation se lance par un ' CALL 2D1H ' avec:
- HL pointant dans la RAM l'adresse ou est stocke le nom du programme (cf WRITE)
- A contenant le type de programme: #30

Cette routine lance le moteur du cassette

b/ Ecriture du texbuffer

Cette routine se lance par un ' CALL 2C8H ' avec :
- HL charge par un ' LHLD 29FH ':debut du bloc a ecrire
- DE contenant le nombre d'octets du textbuffer (cf le calcul de l'exemple)

c/ Ecriture de la table des symboles

Cette routine est la meme que la precedente avec:
- HL charge par un ' LHLD 2A1H '
- DE contenant le nombre d'octets de la table des symboles (cf l'exemple)

Elle est egalement appelee par un ' CALL 2C8H '.

d/ Fermeture du fichier

Cette routine se lance par un ' CALL 2CBH '.Elle stoppe le moteur du cassete.

exemple: la sauvegarde d'un programme basic s'appelant ESSAI

```
        ORG     0A000H
;
        MVI A   30H         ;
        LXI H   NAME        ; OUVERTURE DU FICHIER ET ECRITURE DU NOM
        CALL    2C5H        ;
;
        LHLD    29FH        ;
        XCHG                ; DE CONTIENT L'ADRESSE DE DEBUT DU TEXTBUFER
```

```
        LHLD    2A1H        ; HL CONTIENT L'ADRESSE DE DEBUT DE LA SYMBOLTABLE
        CALL    0DE1AH      ; (HL)-(DE) DANS HL ET (DE) PRESERVE
        XCHG                ;
        CALL    2C8H        ; ECRITURE DU BLOC
;
        LHLD    2A1H        ;
        XCHG                ;
        LHLD    2A3H        ; IDEM QUE PRECEDEMMENT
        CALL    0DE1AH      ; POUR LA TABLE DES SYMBOLES
        XCHG                ;
        CALL    2C8H        ;
;
        CALL    2CBH        ; FERMETURE DU FICHIER
;
        RET
NAME    DB      5H
        DB      'ESSAI'
        END
```

CONCLUSION
****

Je pense avoir ete clair, ce qui n'est pas evident, croyez le.
Cet article est tres important car un chapitre consacre a la deprotection de programmes s'appuie sur ce que vous venez, je l'espere , de lire. J'ai donc tenu a le reecrire comme je l'ai indique plus haut. La deprotection d'un programme connu n'aura plus de secret pour vous quand vous aurez lu les quelques lignes que j'ai ecrites et qui paraitront , je le souhaite, sans problemes.
J'espere pouvoir ecrire prochainement sur le heap pointer et sur les commandes SPL. Peut etre en aurai-je le temps. Si toutefois un DAIiste averti pouvait expliquer clairement (peut courant helas), ces notions ...

LEGRY HP & L.

Pour les DAI-istes cinéphiles (et meme pour tous les amateurs d'ordinateurs), voici un film qui devrait vous intéresser :

Electric Dream

En bref, c'est l'histoire désopilante d'un homme qui achète un ordinateur particulièrement intelligent ... (qui n'est pourtant pas un DAI !). Par la suite, cet homme tombe amoureux d'une femme, et c'est là que l'ordinateur devient jaloux (ou plutot genant !). Il fera tout pour gener son maitre... comment cela va-t il se terminer ???
(à ne pas manquer , sortie depuis peu à Bruxelles et à Paris)

Marc Vandermeersch, IDC association

HARDWARE

STABILISATION DE L'IMAGE

Article original: DAINAMIC GERMANY (RFA)
Traduction: Andres KURT - BUTGENBACH (B)

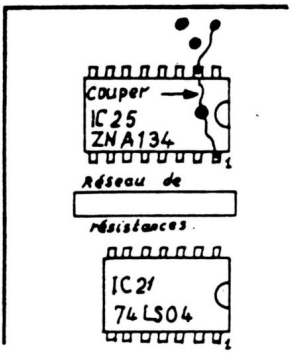
Matériel utilisé:

- fer à souder 40W
- pompe à désouder
- un couteau bien aiguisé
- du fil isolé
- une résistance de 33Kohms
- un condensateur de 1nF5
- une diode 1N4148 ou 1N914

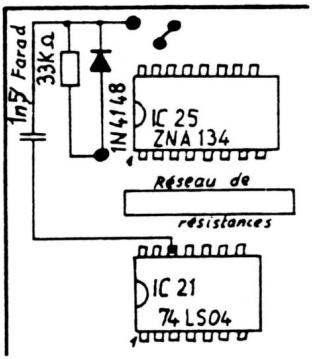
Cette modification n'est possible que sur les machines de révision 3, 4 et 5. Veillez à ce que les soudures soient bien faites et que les pistes du circuit imprimé soient bien coupées.

Travail à exécuter:

- couper la piste du circuit imprimé suivant le schéma.
- souder le condensateur entre la pin 12 de l'IC 21 et le point de soudure à coté de l'IC 25 (à coté de la liaison entre les 2 points de soudure).
- mise en parallèle de la résistance et de la diode entre la pin 1 de l'IC 25 et du point de soudure.



Côté soudures



Côté composants